Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859




# Sensor Data Protection through Integration of Blockchain and Camouflaged Encryption in Cyber-Physical Manufacturing Systems

Zhangyue Shi[1], Boris Oskolkov[1], Wenmeng Tian[2], Chen Kan[3], and Chenang Liu[1]*
[1]The School of Industrial Engineering & Management, Oklahoma State University, Stillwater, OK
[2]Department of Industrial and Systems Engineering, Mississippi State University, Mississippi State, MS
[3]Department of Industrial, Manufacturing, and Systems Engineering, The University of Texas at Arlington, TX
*Corresponding author. chenang.liu@okstate.edu

**Abstract**: The advancement of sensing technology enables efficient data collection from manufacturing systems for monitoring and control. Furthermore, with the rapid development of the Internet of Things (IoT) and information technologies, more and more manufacturing systems become cyber-enabled, facilitating real-time data sharing and information exchange, which significantly improves the flexibility and efficiency of manufacturing systems. However, the cyber-enabled environment may pose the collected sensor data under high risks of cyber-physical attacks during the data and information sharing. Specifically, cyber-physical attacks could target the manufacturing process and/or the data transmission process to maliciously tamper the sensor data, resulting in false alarms or failures in anomaly detection in monitoring. In addition, the cyber-physical attacks may also enable illegal data access without authorization and cause the leakage of key product/process information. Therefore, it becomes critical to develop an effective approach to protect data from these attacks so that the cyber-physical security of the manufacturing systems could be assured in the cyber-enabled environment. To achieve this goal, this paper proposes an integrative blockchain-enabled data protection method by leveraging camouflaged asymmetry encryption. A real-world case study that protects cyber-physical security of collected sensor data in additive manufacturing is presented to demonstrate the effectiveness of the proposed method. The results demonstrate that malicious tampering could be detected in a relatively short time (less than 0.05ms) and the risk of unauthorized data access is significantly reduced as well.

**Keywords**: blockchain; camouflaged encryption; cyber-physical security; manufacturing system; sensor data protection

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


## 1 Introduction

Advanced sensing and information technologies have been increasingly incorporated in the daily operations of manufacturing systems, making them more and more cyber-enabled. For example, a large variety of sensors can be utilized for in-process data acquisition. These collected data contain fruitful information, enabling real-time decision-making regarding quality assurance and process improvement such as *in-situ* process monitoring and real-time control. As another perspective of cyber-enabled manufacturing, cloud-based data storage becomes more and more popular. However, as the manufacturing environment becomes increasingly cyber-enabled, the risk of cyber-physical attacks also increases significantly, which may result in great loss to the enterprises [1, 2].

Recently, there are several studies about cyber-physical vulnerability assessment in manufacturing. For example, the part design files (such as the STL files in additive manufacturing) could be breached in a cyber-enabled environment [3, 4]. Similarly, the collected sensor data may also be altered by cyber-physical attacks. As shown in FIGURE 1, two common types of cyber-physical attacks may occur in a cyber-enabled manufacturing system. First, the malicious tampering could maliciously modify the sensor data. As a result, it may lead to either false alarms or missed detections of anomalies, which could result in enormous time loss and costs to enterprises. Also, malicious modification on sensor data may heavily deteriorate the performance of data analytics methods. Another type of cyber-physical attacks, i.e., unauthorized access, refers to that adversary may illegally access the data. This unauthorized data access may lead to key information leakage and even illegal counterfeiting.

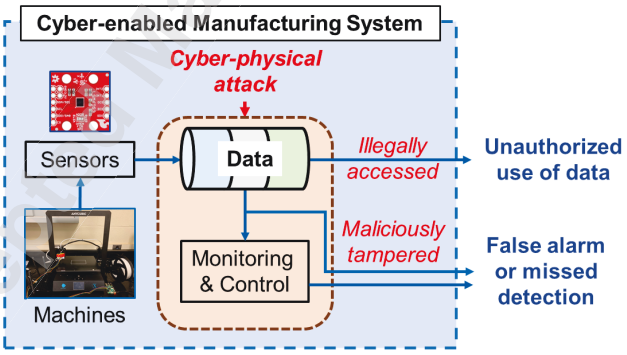


***FIGURE 1:*** *Potential cyber-physical attacks for data in manufacturing systems.*

To improve the cyber-physical security in cyber-enabled manufacturing, recent studies have developed effective data-driven methods such as neural networks for cyber-physical attack detection using sensor data [5, 6]. However, methodologies to prevent the sensor data from unintended modification and unauthorized access in cyber-physical

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


manufacturing are still very limited. In fact, if the sensor data were attacked, the important samples could be replaced or the data distribution could be altered, and thus the performance of those abovementioned data-driven detection methods will be significantly compromised. Therefore, the objective of this study is to develop an effective approach to protect the cyber-physical security of sensor data. There are three major challenges to achieve this goal: (1) the format of sensor data is relatively simple and fixed, which can be easily modified by cyber-attacks in a relatively short time; (2) small changes are difficult to be detected while they could lead to serious product quality issues; and (3) the code-book needs to be updated frequently when using the-state-of-the-art symmetric encryption methods, leading to comparably high maintenance costs.

To address these challenges, this study develops a novel blockchain-enabled approach for sensor data protection in advanced manufacturing systems, which integrates the powerful blockchain and a camouflaged asymmetry encryption framework. The proposed method is able to improve resistance against two cyber-physical attacks (i.e., malicious tampering and unauthorized access) and hence reduces the potential risk of these attacks. Blockchain is a newly developed popular technology that has been applied in a wide range of areas such as cryptocurrencies, supply chain, and smart contracts [7]. It has high resistance against data modification due to its unique structure design. The data stored in the blockchain cannot be altered unless all subsequent blocks are modified. Based on the vanilla blockchain, an engineering-driven blockchain is proposed in this study to accommodate the manufacturing settings. Meanwhile, the proposed camouflaged asymmetry encryption can effectively encrypt the sensor data to prevent unauthorized access and convert the ciphertext to a format similar to the original data, which further reduces the potential attack risks.

Specifically, this work is based on the hypothesis that the encryption-only approach may not be able to provide sufficient security guarantee to cyber-physical manufacturing systems [8, 9]. Therefore, we propose a new data obfuscation/camouflage approach to potentially confuse/mislead the attackers (and thus reduce the likelihood of attack attempts) or possibly slow down the unauthorized access procedure. Another key contribution of this work is the novel integration of blockchain, asymmetric encryption, and data obfuscation, which holistically considers the prevention of malicious tampering and unauthorized access, as well as the attacker's intention. Besides, the proposed methodology also takes the specific domain knowledge of cyber-physical manufacturing systems into consideration. Thus, this work provides a new direction to leverage the blockchain for protecting the security of important process data in cyber manufacturing systems.

The rest of this paper is structured as follows. A brief review of the related research from literature is provided in Sec. 2. The proposed research methodology is elaborated in Sec. 3. Subsequently, Sec. 4 further demonstrates the effectiveness of the proposed method based on a real-world case study. Finally, conclusions and future work are discussed in Sec. 5.

## 2 Literature Review

The study is motivated by the concerns of cyber-physical security for sensor data in advanced manufacturing systems. Thus, this section first briefly reviews the existing studies related to cyber-physical security protection in manufacturing and discusses their limitations (Sec. 2.1). Then, the existing applications of blockchain in manufacturing systems are reviewed in Sec. 2.2. Meanwhile, the research gaps are also identified.

### 2.1 *In-situ* and post-manufacturing cyber-physical security protection in manufacturing

Malicious design/process modification (such as the design geometry, machine parameters, or *in-situ* data modification) may lead to a manufacturing system halt (e.g. false alarm) or quality deterioration (e.g. missed detection of anomalies). Additionally, unauthorized design/data access may result in key information leakage [10]. The risk of these attacks needs to be eliminated at any stage in manufacturing, including the design phase, manufacturing phase, and post-manufacturing phase [4, 11]. This study focuses on the cyber-physical security of the sensor data, which contains both manufacturing and post-manufacturing phases. For the cyber-physical security protection of both phases, sensor data play a significant role in cyber-physical attack detection [12]. Heterogenous sensor signals such as acceleration, temperature, and acoustic emission are common choices for process monitoring [13-16]. In addition, advanced imaging technologies have been developed, providing rich process information. Optical camera, infrared imaging, video, and 3D scan could generate high-dimensional data for process quality control, and now are widely applied in manufacturing systems [17-21].

Correspondingly, data-driven analytics based on the sensor data become popular to detect cyber-physical attacks and improve system resilience [20, 22, 23], which consists of both machine learning methods and statistical methods. In terms of machine learning applications, both supervised and unsupervised monitoring become increasingly adopted. For example, Shi *et al.* developed an autoencoder-based approach to extract features from high dimensional sensor signals for online process monitoring [15]. Li *et al.* incorporated several machine learning algorithms to detect geometry defects at post-manufacturing stage [24]. Another direction is to improve the statistical quality control tools

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


(e.g., control chart), making it applicable for cyber-physical attack detection. For example, Elhabashy *et al.* introduced randomness into control chart to make it more sensitive to cyber-physical attacks [25, 26]. However, currently there are limited studies investigating how to protect the cyber-physical security from the data perspective. Current methods are based on the premise that all data are well protected while it is possible that the data have already been maliciously modified. If the sensor data were already attacked, these methods will not work well or even provide misleading results. Therefore, there is an urgent need in developing an effective approach to detect malicious tampering on steam data during manufacturing.

Some physical-based cyber-physical security detection methods have been proposed in recent years. For the detection of malicious tampering, recent studies have applied sensing techniques such as chemical taggants [27], impedance analysis [28], and physical hash [5] for product authentication, which could cause extra time and material cost. For unauthorized access, there are also several recent studies investigating how to manage and share data [29]. For example, Yen *et al.* [30] proposed a SaaS-centered framework for manufacturing system health management, which facilitates reuse and sharing of sensor data. However, these approaches do not have sufficient capability to ensure the data security. Even though the sensor data can be encrypted and protected by passwords (i.e., symmetric encryption), the data security still cannot be well ensured when the network security is breached. In addition, the code-book needs to be updated frequently for most of common symmetric encryption approaches, causing high maintenance cost [31]. To address these research gaps in the cyber-physical security assurance of advanced manufacturing systems, as a newly developed technology, blockchain-based approaches have demonstrated their great potential. The existing applications of blockchain in manufacturing systems are briefly reviewed in Sec. 2.2.

## 2.2 Applications of blockchain in manufacturing

In recent years, blockchain has been successfully applied to manufacturing systems for different objectives, such as supply chain management, and quality control [32, 33]. For supply chain management, due to its distributed ledge property, blockchain has been adopted in the manufacturing supply chain management, especially in the additive manufacturing which brings high flexibility and is highly distributed [33]. In addition, blockchain has also been applied to decentralized manufacturing systems for data sharing and information processing in the recent studies. For example, Christidis *et al.* [34] applied blockchain to address scalability and security challenges in Internet of Manufacturing Things. Ghuli *et al.* [35] proposed a decentralized system for peer-to-peer identification of ownership

of IoT devices in cloud, which is able to transfer ownership among users without involvement of third parties. Bahga *et al.* [36] proposed a decentralized, peer-to-peer platform for industrial IoT based on the blockchain technology. This method incorporates the digital information components in IoT-based manufacturing to blockchain and enables the participants in a decentralized, trustless, peer-to-peer network to interact with each other without extra cost of a trusted third party. Furthermore, blockchain were further applied in data sharing and transactions recording at the enterprise level. For instance, Yu *et al*. [37] constructed a blockchain-based structure to enhance the information transparency and decentralization in cloud manufacturing, in which the smart contracts were applied to deal with manufacturing services in cloud platform. Shafagh *et al.* [38] designed a blockchain-based system for IoT, which brings distributed access control and data management.

Although blockchain has been increasingly applied to manufacturing, most existing studies are focused on the macro-scale enterprise level activities in decentralized manufacturing systems, such as anti-counterfeiting and information sharing. In manufacturing, in addition to data collected during manufacturing process, the fabricated part itself could become an important data source for organizations [33]. Certification and quality assurances need to be implemented for the whole manufacturing processes. The digital representation of a product and its corresponding data can be seen as a digital twin [39]. Blockchain's ability to manage the ownership of data has the potential to protect cyber-physical security of digital twin data of fabricated products such as G-code and sensor data. For example, Kennedy *et al.* [40] incorporated a QR code with a 3D printed part, in which the designed features are included, and further forms a digital twin of the physical part in blockchain to improve the product security. In the prior work of the authors, blockchain was successfully applied to G-code protection in additive manufacturing [10]. Compared with the sensor data protection, the number of G-code is fixed after slicing while the sensor data are collected dynamically. Besides, ciphertext may cause more malicious decryption attempts from the adversaries, which increases the potential risk as well. Consequently, our prior work is not sufficient to protect the online stream data [10], which motivates this study to further extend it and make it more suitable for sensor data protection.

## 3 Proposed Research Methodology

To prevent the *in-situ* sensor data from malicious tampering and unauthorized access in advanced manufacturing systems, the overall framework of the proposed blockchain-enabled methodology is composed of the following three aspects:

1) A data storage approach using blockchain: In Sec. 3.1, a blockchain-enabled approach is proposed to store sensor data, which is able to detect malicious tampering on data fast.

2) A camouflaged asymmetry encryption framework: In Sec. 3.2, a camouflaged asymmetry encryption framework is developed to further reduce the risk of unauthorized data access.

3) Integration of the proposed method for sensor data protection in manufacturing: In Sec. 3.3, integration of the proposed method to protect sensor data from both malicious tampering and unauthorized access is elaborated.

### 3.1 Blockchain-enabled sensor data storage

To prevent the malicious sensor data modification, a blockchain-enabled sensor data storage approach is first proposed in this section. Blockchain provides a safe and trustworthy platform for peer-to-peer communication, which could be used to store a variety of important trackable information, such as healthcare data and transaction records [41]. One notable feature of blockchain is the incorporation of hash cryptography, which contributes a lot to assuring cyber-security. In hash cryptography, the hash function is a one-way function that maps data to a fixed-size hash value [42], and it is impossible to reversely derive the original contents from the generated hash value. Specifically, there are two critical properties of hash function to ensure data security. First, this is a one-to-one mapping, i.e., if two inputs $x_1$ and $x_2$ are different, their generated hash values must be different. Second, the hash function is a non-invertible function. Given a hash value, the original input text cannot be derived. In practice, the commonly used hash functions include secure hash algorithms such as SHA-2, which takes text as input and output a hexadecimal string [43].

Block header and block body are two major components of blockchain. The important file/data are stored in block body and the unique identification information of each block is stored in block header. As shown in FIGURE 2, block header contains the following items to ensure the uniqueness and security of block:

1) Hash of the previous block: a hash value representing the previous block.

2) Hash of the current block: a hash value representing the current block, which can be calculated from hash of previous block, current block index, timestamp and the data stored in the current block.

3) Timestamp: current timestamp in second format.

With a unique cryptographic hash identification, each block is chained with its neighboring block via the hash of the previous block. The data in the blockchain are strictly ordered since the latter block cannot be connected to the

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


chain without hash value of previous block. Besides, due to the uniqueness of hash function, any modifications on the stored data will lead to a completely different hash value, which could be detected quickly and accurately (as demonstrated in Sec. 4). Therefore, leveraging blockchain for data storage could prevent malicious tampering on sensor data because of its capability to detect even a very slight unintended modification.

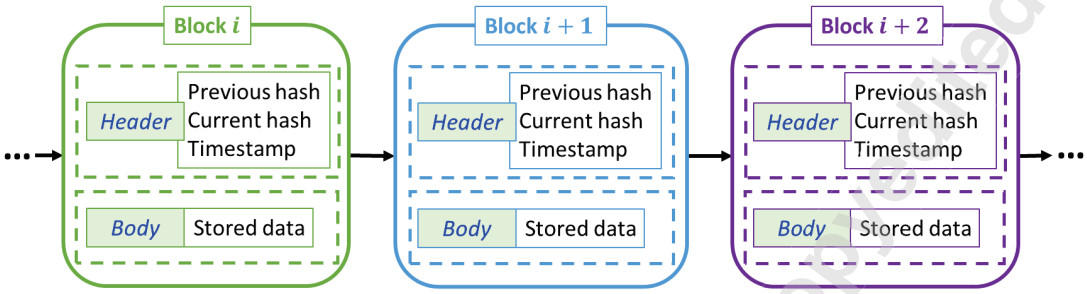


***FIGURE 2:*** *A demonstration of the blockchain structure.*

Similar to blockchain, the sensor data are collected in a sequential ordered as well. As demonstrated in FIGURE 3, the sensor data collected in each time window can be treated as the data stored in one block which connects the previous block (i.e., the previous time window) through the hash value. When storing sensor data in block, a corresponding unique hash value of the current block could be generated. Any unintended modifications on the collected data will result in a significant change of hash value in the corresponding block due to the uniqueness property of hash. If adversary attempts to tamper the sensor data to further manipulate the manufacturing process or deteriorate product quality, it could be detected accurately through mismatch of hash value. Besides, storing sensor data in blockchain also enables users to locate the exact modification on sensor data in a timely manner (see demonstrations in Sec. 4).

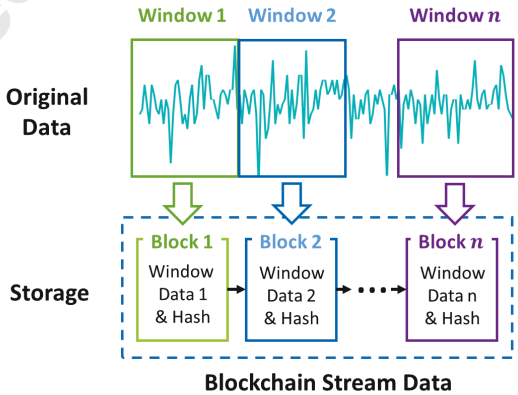


***FIGURE 3:*** *Sensor data storage based on the proposed blockchain architecture.*

It is worth mentioning that there are several differences between the blockchain-enabled structure proposed in this paper and the conventional blockchain. On the one hand, the goal of this study is to protect sensor data collected in a designated manufacturing system. Therefore, the collected data should be stored by the manufacturer so that distributed ledger is not incorporated in this work. On the other hand, in a conventional blockchain, mining is a powerful feature which keeps adding new blocks to the end of chain after proof of work [7]. However, sensor data are collected through designated sensors. Consequently, the new block could only be added through them. Hence, the proof of work is also removed from mining mechanism in this work.

In summary, the proposed blockchain-enabled sensor data storage approach is capable of detecting and locating malicious tampering in a timely manner, which significantly enhances the resistance against malicious tampering. In Sec. 3.2, the camouflaged asymmetry encryption framework is incorporated in the blockchain to further enhance the robustness against unauthorized access.

### 3.2 Camouflaged asymmetry encryption framework in blockchain

Malicious tampering could be detected and located by storing sensor data in blockchain. In conventional blockchain applications, the data stored in blockchain are open and accessible for every user. However, in manufacturing, sensor data contains a large amount of valuable information and some of them may be confidential. Making the data open access may leak key information and result in irreversible loss. Hence, these important data should not be open to the public except for the data owners and users. Directly storing sensor data in blockchain without encryption may result in another type of cyber-physical attack, i.e., unauthorized data access. Consequently, necessary encryption technology should be incorporated so that the information could only be accessed by the designated users. The process of encryption involves manipulating the plaintext using a set of rules or mathematical functions that transform it into ciphertext. Ciphertext is intended to protect sensitive information from unauthorized access or disclosure by making it unreadable to anyone who does not possess the appropriate key or decryption algorithm. It is a critical aspect of modern information security systems, including secure communication channels, digital signatures, and data storage. Meanwhile, ciphertext stored in block contains letters and symbols, indicating that these data are encrypted. This may further cause more malicious attempts on decryption and increase the risk of information leakage. Thus, encrypting important data and reducing the attempts of adversary on ciphertext decryption are two important goals in this study, which are achieved by developing the pluggable options of camouflaged

encryption. Specifically, the encryption technique is applied to keep the data confidential, and the camouflage technique (i.e., the invertible transformation on the ciphertext) is added to reduce the risk that adversary decrypts ciphertext, which are presented in Sec. 3.2.1 and Sec. 3.2.2, respectively.

### *3.2.1 RSA asymmetry encryption framework*

The cryptography approaches are composed of symmetry and asymmetry methods. In general, asymmetry encryption approaches do not need time synchronization among users and do not require secure channel between sender and recipient. Conversely, symmetric encryption approaches require a secure channel. As the symmetric methods utilize the same key for both encryption and decryption. If the key gets attacked, attacker could easily obtain all important information in the entire system. In addition, asymmetric encryption enables the recipient to verify and authenticate the message's source, making it easier to avoid encrypted messages from unknown senders. Compared to the symmetric methods, asymmetric methods have two keys to implement encryption and decryption tasks, respectively. They do not need time synchronization among users and are less vulnerable to cyber-physical attacks [44]. Hence, the asymmetric approach is adopted in this study, which consists of two different keys, i.e., encryption key and decryption key. The paired use of encryption and decryption keys makes it effective to reduce the risk of information leakage. The working principle of asymmetry encryption is simple: A encryption key is used to encrypt the sensor data to the ciphertext and a decryption key is used to decrypt the ciphertext to the original data.

In practice, Rivest-Shamir-Adleman (RSA) is a widely used asymmetry encryption approach due to its great efficiency. The Digital Signature Standard (FIPS 186-5) [45] defines the acceptable level of how the RSA key generation procedure can ensure the system solidness, with the specific key generation procedure, including padding. Thus, this study follows this standard for demonstration purposes. Notably, it is also possible that RSA is not good enough due to inappropriate application domain, hardware platform, or other factors.

Mathematically, the process could be formulated as,

$$\boldsymbol{y} = f(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ denotes the original text and $\boldsymbol{y}$ denotes the ciphertext. $f(\cdot)$ denotes the encryption key. Then the decryption could be formulated as,

$$\boldsymbol{x} = g(\boldsymbol{y}) \tag{2}$$

where $g(\cdot)$ denotes the decryption key. Notably, $g(\cdot)$ could derive $f(\cdot)$ and this derivation from $g(\cdot)$ to $f(\cdot)$ cannot be inverted. In other words, given $g(\cdot)$, $f(\cdot)$ can be derived while $g(\cdot)$ cannot be derived from $f(\cdot)$. Before storing into a block, the collected sensor data are encrypted to ciphertext using the encryption key first, which makes them only accessible to the designated agents.

More specifically, in the RSA cryptosystem, $f(\cdot)$ could be presented as:

$$f(\cdot) = m^e \pmod{n} \tag{3}$$

and for $g(\cdot)$:

$$g(\cdot) = (m^e)^d \pmod{n} \tag{4}$$

where $m$ is the original text, $e$ is the encryption key value, $n$ is modulus size, and $d$ is decryption key value. More details about RSA are presented in the literature [46].

In this study, for the RSA keys generation, we have used the PyCryptodome package in Python. The algorithm closely follows FIPS 186-5 in its sections B.3.1 and B.3.3 [45]. The modulus is the product of two non-strong probable primes and its size chosen as 1,024 bits. Each prime passes a suitable number of Miller-Rabin tests with random bases as well as a single Lucas test. In this study, according to the abovementioned literature, the security level corresponds 80 “bits of security”, as we used modulus size equals 1024.

It is also worth noting that the adoption of the RSA asymmetry method in this study is mainly for demonstration purposes. In practice, other common asymmetry encryption methods could also be applied to replace RSA, such as the Elliptic-curve cryptography (ECC) [47]. Specifically, in Mahto *et al.*’s work [48], a performance comparison between RSA and ECC was conducted. This comparison indicates that the least total encryption-decryption time for RSA only exists in low security systems but requires additional enhancing like using Chinese Remainder Theorem or Multi-prime RSA. When the security level (more than 112 “bits of security”) is increased, then the ECC will outperform RSA. Besides, Saho *et al.* [49] identified that ECC could be more suitable for embedded systems as ECC generally requires less computational resources. With the incorporation of asymmetry encryption, subsequently, the camouflage technique can be applied to further reduce potential attempts of adversary to decrypt on ciphertext.

### *3.2.2 Proposed camouflage technique*

After asymmetry encryption, ciphertext $\boldsymbol{y}$ are composed of numbers, letters, and symbols (see a demonstration in Sec. 4.2), indicating that the data are encrypted. This could lead to the situation that the adversary tries decrypting the ciphertext and increases the potential risk of information leakage. As the quantum technology develops, some encryption methods (such as RSA) can be breached [50]. If a hacker obtains the private key, this hacker can decrypt the ciphertext easily.

To address this issue, a natural idea is to consider the data obfuscation. In practice, the common data obfuscation techniques include special storage and encoding, aggregation and different ordering of data [51]. Most of the data obfuscation approaches can be grouped into three categories [52]:

1) **Data randomization** works by perturbing the data, making it difficult to reconstruct the original values and preserves sensitive data.
2) **Data anonymization** applies generalization and suppression to a dataset, where generalization replaces a value with a less specific one, while suppression does not release a value at all.
3) **Data swapping** swaps the values within a single field in a record set. This makes it difficult to match individual records, but it does not affect the overall statistics of the data set [53].

According to the literature, most of the existing obfuscation techniques did not consider the need in making the format of encrypted data consistent with the original data, for example, the collected from an accelerometer sensor in this study. Thus, a special aspect of data obfuscation, namely, a camouflage strategy, is proposed in this study. It is true that the obfuscated/camouflaged data may not be able to mislead the malware attacks. Nevertheless, when an attack is performed by a human, the proposed camouflage/obfuscation approach can reduce the likelihood of attack attempts, as the camouflaged data will look very similar to the original data (as presented in Sec. 4.2), and therefore enhance the security.

The proposed procedure monotonically transfers these ciphertext to numeric format first (see FIGURE 4). Hereafter, the mathematical transformation (e.g., mean shift, and scaling) could be applied to scale the data, making camouflaged data have similar scale compared to the original data. With help of this additional camouflaging technique which masks the ciphertext to original data format, the risk that hacker tries to decrypt the ciphertext could decrease.

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


In addition to that, even if hacker obtains the private key, attacker will retrieve signals in an incorrect space, which cannot be effectively utilized. The transformation is invertible, which could be mathematically formulated as,

$$\tilde{\boldsymbol{y}} = h(\boldsymbol{y}) \tag{4}$$

where $h(\cdot)$ is the monotonically reversible transformation function, which could consist of but not limited to binary-ASCII transformation, string-number transformation, digit split, and scaling.

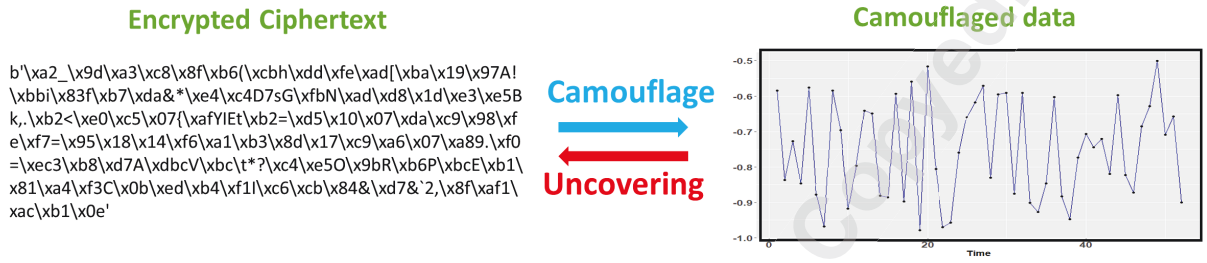


***FIGURE 4**: A demonstration of the camouflaged encryption.*

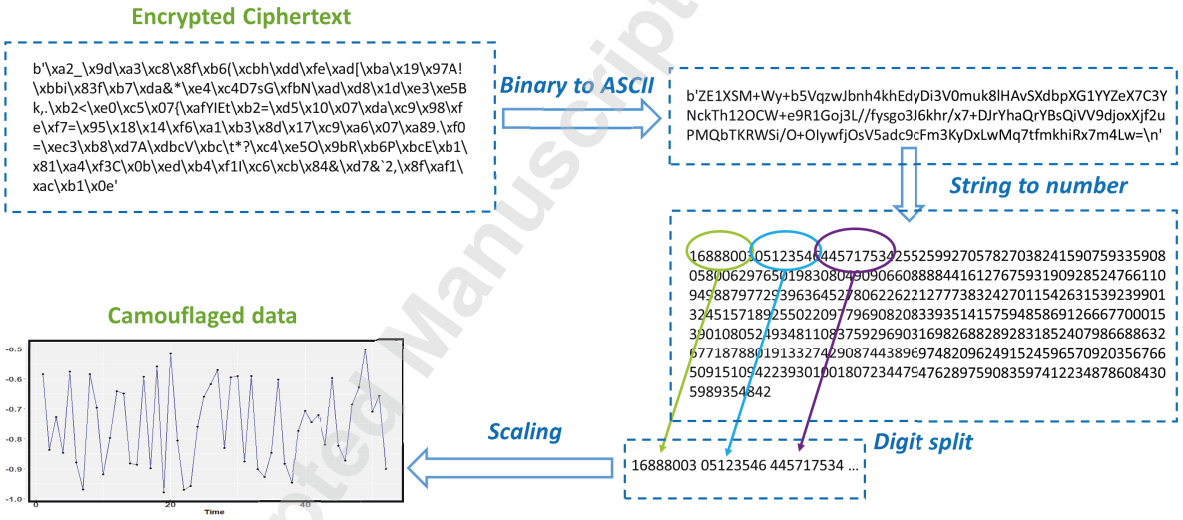


***FIGURE 5:** The detailed procedures of the proposed camouflage framework.*

The detailed camouflage process is displayed in FIGURE 5. After RSA encryption, the ciphertext is in binary format. Binary-ASCII transformation could transfer the binary ciphertext from binary format to ASCII strings, which is helpful for the follow-up camouflage processing. The string-number transformation is capable of mapping string to number monotonically. Hereafter, numerical format ciphertext split into digits with equal length. Finally, scaling is

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


performed on these equal-length digits to make them have similar format and value with original data. Scaling is one necessary step of camouflaging since the camouflaged data may not be in the same scale as the original sensor data after digit split. Attacker with engineering domain knowledge immediately knows that the data have been encrypted. Therefore, we need to further scale them to a similar scaling level and make hackers believe the data have not been encrypted, which could further reduce the attempt from attackers to decrypt the ciphertext. In addition to these steps, any reasonable invertible transformation could be added as well.

After camouflage, $\tilde{\boldsymbol{y}}$ has similar format with the original data $\boldsymbol{x}$. The camouflaged data $\tilde{\boldsymbol{y}}$ could be uncovered to ciphertext $\boldsymbol{y}$ using the inverse function $h^{-1}(\cdot)$. With help of camouflage, adversary does not know that the stored data are encrypted, and the risk of information leakage is further decreased. As displayed in FIGURE *6*, Before storing sensor data into block, camouflaged asymmetry encryption method is applied to effectively prevent sensor data from unauthorized access.

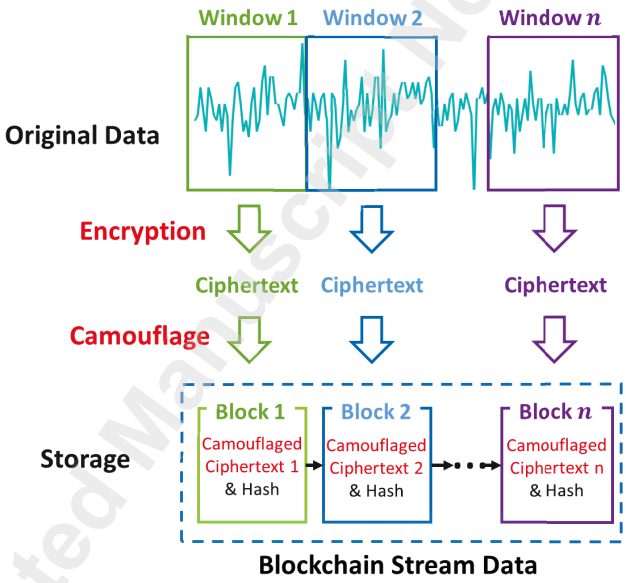


***FIGURE 6:*** *Overview of the blockchain-based camouflaged asymmetry encryption storage for sensor data.*

In summary, the steps of sensor data encryption, camouflage and sharing are displayed in FIGURE *7*, which contains three parts: (1) decryption and encryption key generation; (2) sensor data encryption and camouflage; and (3) uncovering and decryption. Before encryption, data user generates two keys: encryption key and decryption key. According to RSA, decryption key $g(\cdot)$ is generated first. Afterwards, encryption key $f(\cdot)$ is derived from $g(\cdot)$ and the derivation is irreversible. The ciphertext encrypted by $f(\cdot)$ can only be decrypted by $g(\cdot)$, ensuring the data

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


could only be accessed by the user. After acquiring encryption key by the manufacturer sensor data $\boldsymbol{x}$ are encrypted to ciphertext $\boldsymbol{y}$ and camouflaged to form $\tilde{\boldsymbol{y}}$ which has similar form with $\boldsymbol{x}$. In practice, the data owner offers designated users blockchain-stored data and uncovering method $h^{-1}(\cdot)$. Notably, the decryption key $g(\cdot)$ and uncovering method $h^{-1}(\cdot)$ work separately to uncover $\tilde{\boldsymbol{y}}$ and decrypt $\boldsymbol{y}$. The camouflaged encrypted sensor data are only accessible for authorized users who own decryption key $g(\cdot)$ and know the camouflage method $h^{-1}(\cdot)$, which decreases the risk of critical information leakage.

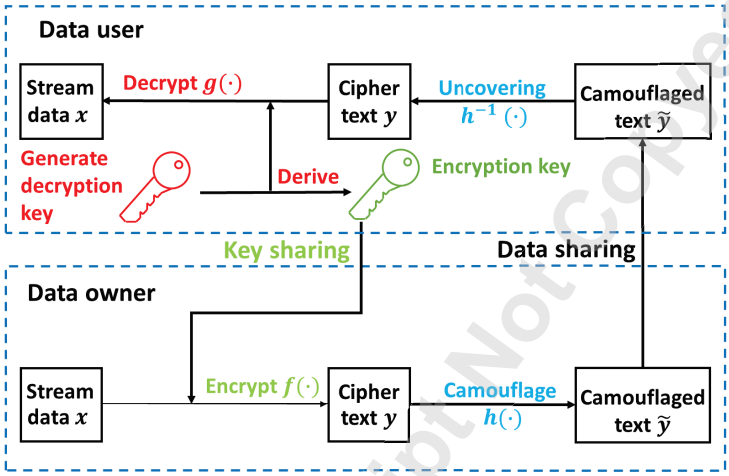


***FIGURE 7:*** *Steps of sensor data encryption, camouflage, and sharing.*

With the application of the camouflaged encryption framework, it is challenging to know the appropriate uncovering method and very time-consuming to decrypt the ciphertext of even a single block. Therefore, it becomes difficult to obtain the original data in a short time since the number of blocks may be large. Incorporating the camouflaged asymmetry encryption method in blockchain storage structure, the proposed approach is capable of resisting the malicious tampering and unauthorized access, which is discussed in Sec. 3.3.

### 3.3 Integration of the proposed blockchain-enabled data protection approach

In practice, manufacturers store their collected data locally or on the cloud. The manufacturers could frequently verify the data integrity in order to detect malicious tampering in a timely manner. The paradigm of the proposed blockchain-enabled framework with camouflaged encryption is illustrated in FIGURE 8, which consists of four steps.

- **Step 1: Key generation**. The data owner generates a decryption key and derives encryption key from the decryption key.

- **Step 2: Sensor data collection**. The sensor data are collected and organized in a window-based format.
- **Step 3: Data encryption, camouflage, and storage**. The collected data are encrypted using the encryption key and then camouflaged. Afterwards, the camouflaged data are stored into blocks. Newly collected data could be continuously added to the end of the chain. The hash of block is recorded and stored in a cyber-disabled environment for verification purpose. The cyber-disabled environment means the environment without Internet access so that attacker cannot modify data in this environment.
- **Step 4: Verification on stored data**. The manufacturer performs frequent inspection on the stored data by recalculating the hash of each block to see whether there is a mismatch on the hash values which indicates occurrence of malicious tampering at the corresponding block.

Using the proposed blockchain-enabled framework, malicious tampering could be effectively prevented. In general, malicious tampering could be categorized into two types, namely, deletion/addition of blocks in the blockchain and slight/severe data modification. According to each type of tampering, there are several verification ways based on the mismatch of hash values. Notably, the verification procedure could be automatically implemented in a relatively short time (See details in Sec. 4), which ensures the sensor data integrity.

For the malicious deletion/addition, there are two approaches to detect it. The first approach is dimension comparison, which directly detects the deletion/addition by comparing current dimension (i.e. the number of blocks) with the expected dimension. In this study, the expected dimension could be determined by the window size, sampling frequency, and manufacturing time. When the dimension of blockchains does not match, it implies occurrence of malicious block deletion/addition. Although dimension comparison is simple and fast, it has several limitations: (1) it cannot locate which block has been maliciously deleted/added, and (2) if the same number of blocks are deleted and added simultaneously, it cannot detect the malicious tampering since the dimension keeps the same. To address these limitations, the second method is developed, namely, chain inspection, which compares the hash value along the chain. FIGURE 9 (a) is a demonstration of malicious deletion detection by chain inspection. The hash value does not match comparing the block $i$'s hash value with the previous hash value of block $i+2$ when malicious deletion occurs. Hence, the deleted block (i.e., block $i+1$) could be detected accurately.

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


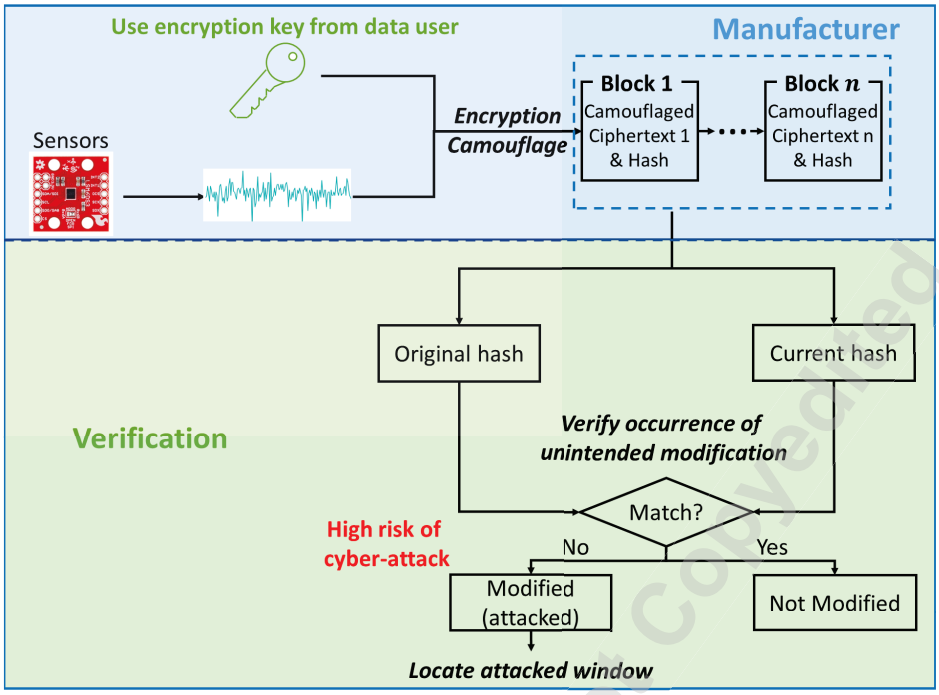


***FIGURE 8:*** *Paradigm to integrate the proposed blockchain-based camouflaged encryption framework in manufacturing systems.*

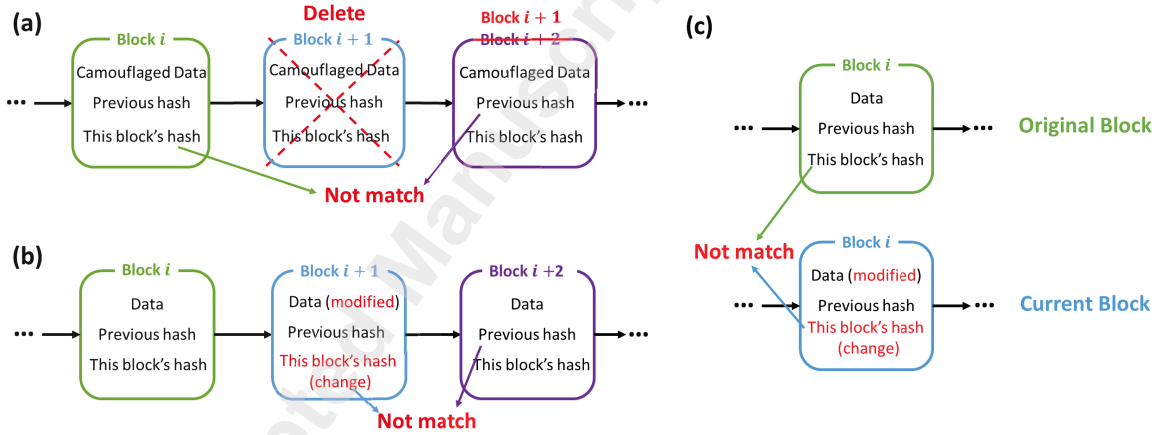


***FIGURE 9:*** *(a) Malicious deletion detection by chain inspection; (b) slight malicious modification detection by chain inspection; and (c) severe malicious modification detection by benchmark comparison.*

In terms of the slight/severe data modification, slight modification refers to the modification on one or several blocks and severe modification refers to the modification starting from certain block till the last block. The slight modification could be detected by chain inspection as well, which is illustrated in FIGURE 9 (b). When the camouflaged data are stored in blocks, the unique hash value for each block is generated. After data modification, an

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


entirely different hash value will be generated during verification. For slight modification which only modifies several specific blocks, the hash value mismatch between the modified and unmodified block denotes the occurrence of the malicious tampering. For example, the data in block $i+1$ are modified by the adversary and the hash value of block $i+1$ changes after recalculating hash value. The previous hash value in block $i+2$ remains unmodified so that by comparing the hash value of block $i+1$ and the previous hash in block $i+2$, the mismatch could be detected accurately. For severe modification, the chain detection does not work since hash values in all the following blocks have been tampered. Thus, benchmark comparison is effective to deal with this problem. Specifically, the original hash value is stored in a cyber-disabled environment and set as benchmark after storing the sensor data into blocks. When doing the hash value comparison, the original hash benchmark is loaded. By comparing the original hash value with current hash value (see FIGURE 9 (c)), the tampered block could be detected in a timely manner.

In addition, with the help of camouflaged asymmetry encryption, it will be very challenging for the adversary to: (1) identify if the data have been encrypted or not; and (2) decrypt the ciphertext in a manageable time [54]. Therefore, the proposed method also significantly reduces the risk of unauthorized access. Notably, the data will be protected using the proposed method once the data are collected. Then the common quality control tools such as control charts or data-driven monitoring methods could be further incorporated without concern on the data correctness. In the post-manufacturing phase, the frequent verification also eliminates the risk of unintended modifications. When an outside user needs to access the data, they could send a request to the manufacturer and provide the encryption key $f(\cdot)$. Subsequently, the manufacturer could securely share the uncovering method $h^{-1}(\cdot)$ with the user so that the user could download the data from the cloud, uncover and decrypt them to the original ones.

The proposed method is an engineering-driven framework which takes several engineering domain knowledge into consideration. First, streaming data are collected in a chronological order and usually are analyzed in a window-based format to effectively utilize the temporal information in practice. In terms of blockchain, each block could store its own data, which highly matches the way of data collection in engineering. In addition, the scaling in camouflage is another perspective to incorporate engineering knowledge. Camouflaged data need to be scaled to appropriate level according to different types of sensors, which is highly correlated with specific engineering applications. To further demonstrate the effectiveness of the proposed method, a real-world case demonstration in additive manufacturing is provided in Sec. 4.

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


## 4 Case Study

This section provides a real-world application of the proposed method based on an additive manufacturing process, i.e., fused filament fabrication (FFF), by protecting the cyber-physical security of the *in-situ* sensor data. The experimental setup and data collection are introduced in Sec. 4.1, the sensor-data encryption and decryption is introduced in Sec. 4.2, and Sec. 4.3 presents the analysis on cyber-physical attack resistance.

**TABLE 1:** The process parameter of designed part.

| Parameters | Value |
|---|---|
| Printing speed | 40 mm/s |
| Layer thickness | 0.3 mm |
| Nozzle temperature | 215 °C |
| Bed temperature | 60 °C |

### 4.1 Experiment setup and data collection

In this study, a desktop FFF 3D printer was used for data collection. To collect sensor data during manufacturing, a vibration sensor (i.e., MEMS accelerometer) was installed on the printing bed, which could collect real-time vibrations in three-axis with a sampling frequency of 3 Hz. FIGURE 10 displays the FFF printer and sensor installation [6]. ARDUINO MEGA 2560 REV3 microcontroller was used for data collection from the sensor. In this study, a cube with dimension 2cm× 2cm× 2cm was fabricated with the machine using the process parameters shown in TABLE 1. After experimental platform setup, the stream data could be collected. In this study, the window size is set as 10 sample points. Each window is encrypted and camouflaged individually.

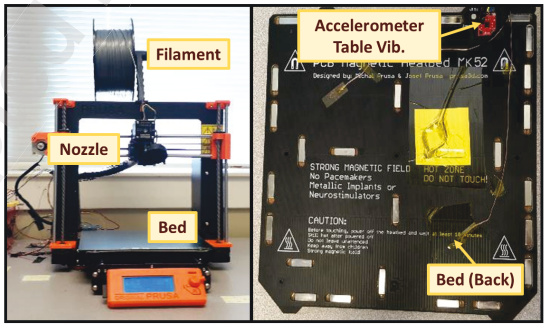


***FIGURE 10:*** *The experimental platform setup.*

### 4.2 Sensor data encryption and camouflage

During the printing process, the online sensor data are collected and organized in a window-based format, and then they are encrypted to ciphertext first. $g(\cdot)$ is the private key generated by python module RSA from the PyCryptodome library. Afterwards, $g(\cdot)$ could derive public key $f(\cdot)$. As demonstrated in FIGURE 11, it can be observed that the ciphertext looks completely different from the original data. Hereafter, the proposed data camouflage approach is applied. In terms of camouflaging function $h(\cdot)$, it consists of several invertible steps: binary to ASCII; string to number; digit split; and scaling. The ciphertext is converted to ASCII string first and then converted from the string to numeric format. Subsequently, the converted numbers are split into different parts and scaled to a similar level with the original data. The camouflaged data may have different sampling frequency (see FIGURE 12) from original data since there are many digits after string converted to number. As shown in TABLE 2, the total time for encryption and camouflage of each window is about 0.43ms, which is significantly lower than the sampling interval 0.33s (i.e., 3Hz sampling frequency). Furthermore, the time to uncover and decrypt for each window data is 3ms, which is also short enough compared with the sensor sampling period. Thus, the computational efficiency is good enough for the application under *in-situ* situation. This study is performed on Intel Core i5-7400 CPU (3.6GHz) under in Python version 3.7.6. For higher frequency needs on the encryption and camouflaging, it could be achieved either using more advanced hardware settings or using smaller window size.

**TABLE 2:** Computation cost for each operation

| Window Size | Encryption & Camouflage time | Uncovering & Decryption time | Sampling Period |
|---|---|---|---|
| 10 | 0.43 ms | 3 ms | 0.33 s |

Afterwards, the camouflaged stream data are stored in a blockchain. As discussed in Sec. 3.1, each block stores one window of stream data and generates a unique hash value, which is illustrated in FIGURE 11 as well. For demonstration purposes, a tiny blockchain class is built up in Python containing index, data, previous hash, and current hash value.

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


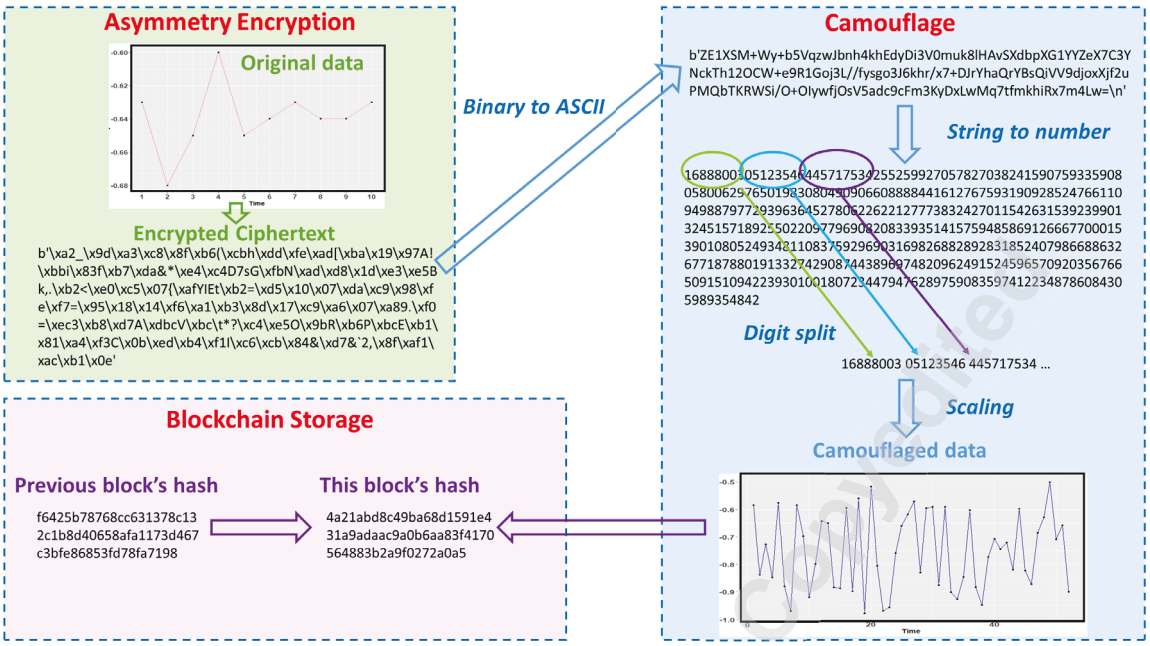


***FIGURE 11:*** *Results demonstration for the camouflaged asymmetry encryption and storage of window-based stream data.*

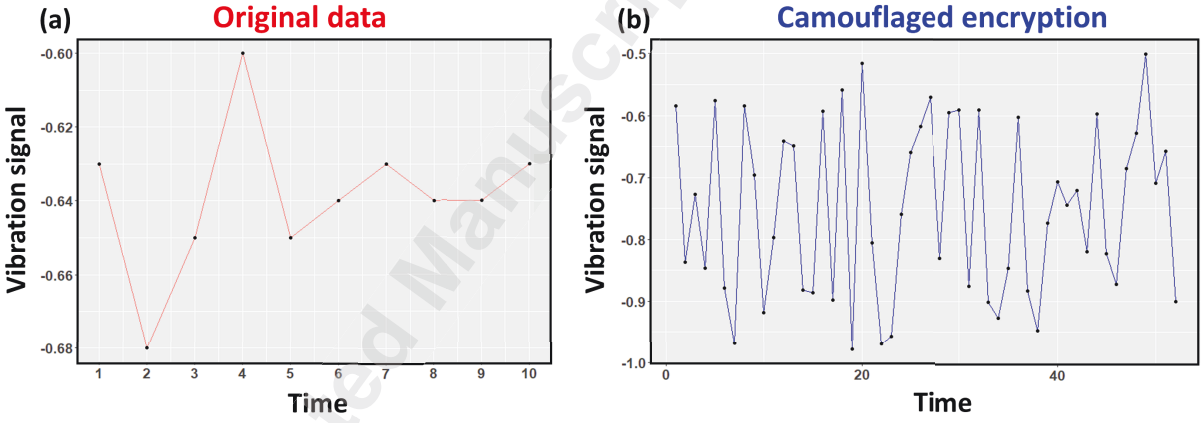


***FIGURE 12:*** *A demonstration of original data (a) and camouflaged data (b) of one window.*

## 4.3 Analysis of cyber-physical attack resistance

By incorporating asymmetry encryption, without decryption key, it will take a very long time to decrypt ciphertext. In addition, the proposed camouflage technique also potentially reduces the risk of decryption attempt. The resistance against unauthorized data access is significantly improved.

Journal of Computing and Information Science in Engineering. Received January 09, 2023;
Accepted manuscript posted October 08, 2023. doi:10.1115/1.4063859


Storing the online sensor stream data in a blockchain makes it more effective to detect the malicious tampering. To detect malicious deletion/addition on blocks, as discussed in Sec. 3.3, dimension comparison and chain inspection could be applied. Dimension comparison is quick and simple, but its capability is limited. To ensure accurate detection, it is necessary to apply chain inspection as well. FIGURE 13 provides a specific demonstration of chain inspection, and this case assumes that block 7 is maliciously deleted. By comparing the current hash in block 6 with the hash of previous block in block 8, the mismatch could be detected quickly. For each window, chain inspection only takes 0.02ms, which is also applicable for the *in-situ* situation.

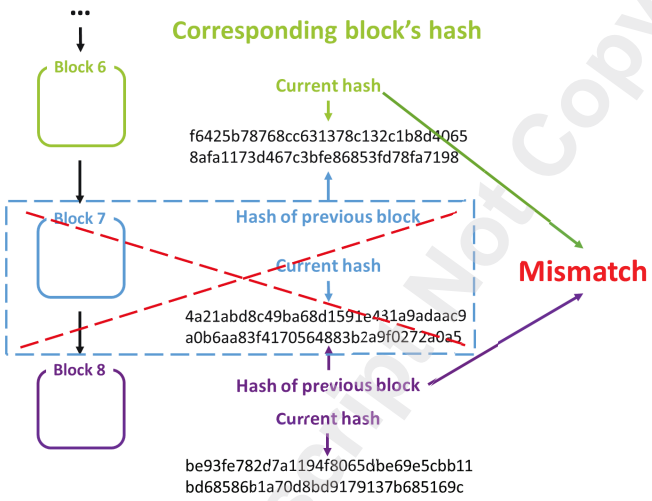


***FIGURE 13:*** *Malicious deletion detection by chain inspection.*

In addition, slight data modification could be detected by the chain detection as well. In the case study, we maliciously modified the first digit in block 7 from 1 to 2. After recalculating hash value of block 7, the hash value became totally different as shown in FIGURE 14. Afterwards, we tried to use chain detection to detect the modification. The current hash in block 7 and previous hash in block 8 didn't match. Therefore, we can locate the exact modification happening in either block 7. And the computational time is 0.062 ms for each block, which is very fast. With help of chain detection, the mismatch could be located accurately and in a timely manner. However, for the severe malicious modification, the chain detection does not work since all the following blocks are tampered. Therefore, the benchmark comparison is implemented which compares the current hash value with the benchmark hash value (see FIGURE 9 (c)). The benchmark blockchain has been developed once the data are stored into the blockchain, which is stored in an environment where no Internet access. By comparing the hash value of current

blockchain with that in benchmark blockchain, the exact modified block could be located. For example, in this case, the hash value in block 7 of current blockchain does not match with that in benchmark blockchain. Since benchmark comparison needs to compare the current hash value with those in benchmark block, the time cost is higher than chain detection, which takes 0.068 ms for each block but still fast enough.

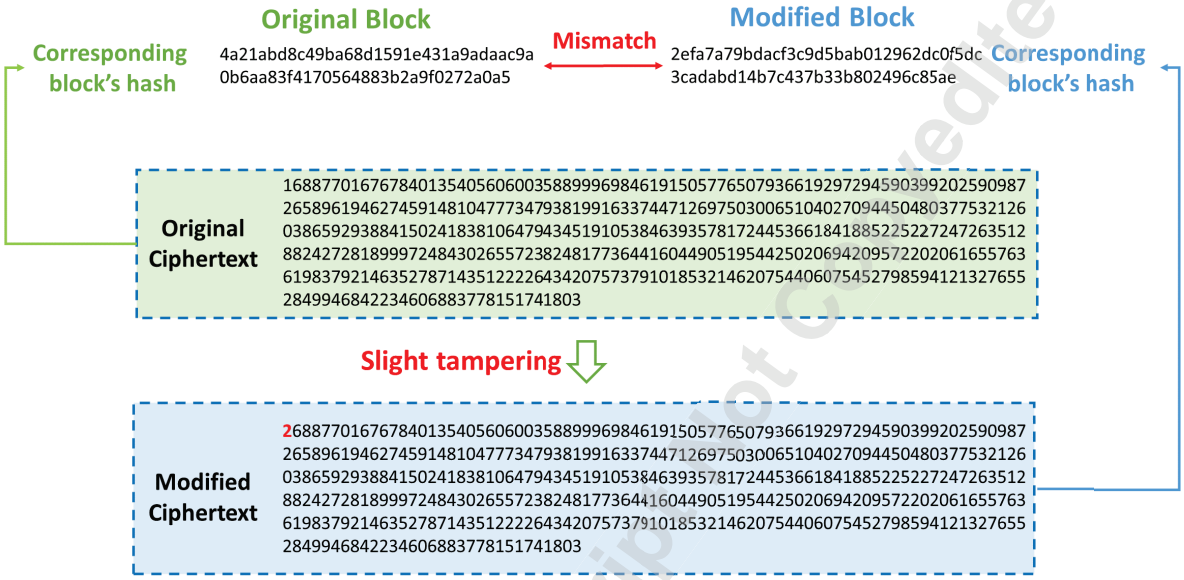


***FIGURE 14:*** *Hash value comparison between before and after slight modification.*

## 5 Conclusions and Future Work

This paper develops a blockchain-enabled methodology to protect the security of sensor data in cyber-enabled advanced manufacturing. Both malicious tampering and unauthorized access of the sensor data could be effectively prevented. Based on the proposed blockchain-enabled data storage, malicious tampering could be detected in accurately and timely via the comparison between hash values. Meanwhile, by incorporating the proposed camouflaged asymmetry encryption method, the risk of unauthorized access could be significantly reduced as well. Furthermore, a preliminary case study in additive manufacturing is conducted to demonstrate the procedure of the sensor data collection, encryption, camouflage, and malicious tampering detection, which also shows that the proposed approach is very promising.

The future work mainly lies in the following three directions. First, exploring other camouflage techniques to mask the ciphertext, which could further reduce the size of camouflaged data and make the storage more effective.

Second, how to protect security of keys and apply other types of asymmetry encryption approaches will be investigated. For example, one of the most common approaches, i.e., encrypting data itself by the symmetrical method and then ciphering the symmetrical encryption key using asymmetrical ways, can potentially be incorporated to the proposed method for further improvement of security protection. Third, more real-world applications will be further explored to examine the effectiveness of the proposed framework.

## Code Availability Statement

The code of this work can be accessed in https://github.com/ShiZhangYue/Blockchain-for-in-situ-data

## Acknowledgement

This work is partially supported by the National Science Foundation under Award Number IIP-2141184.

## REFERENCES

1. Yang, H., S. Kumara, S.T. Bukkapatnam, and F. Tsung, *The internet of things for smart manufacturing: A review.* IISE Transactions, 2019. **51**(11): p. 1190-1216.
2. Chaduvula, S.C., A. Dachowicz, M.J. Atallah, and J.H. Panchal, *Security in cyber-enabled design and manufacturing: A survey.* Journal of Computing and Information Science in Engineering, 2018. **18**(4): p. 040802.
3. DeSmit, Z., A.E. Elhabashy, L.J. Wells, and J.A. Camelio, *An approach to cyber-physical vulnerability assessment for intelligent manufacturing systems.* Journal of Manufacturing Systems, 2017. **43**: p. 339-351.
4. Sturm, L.D., C.B. Williams, J.A. Camelio, J. White, and R. Parker, *Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. STL file with human subjects.* Journal of Manufacturing Systems, 2017. **44**: p. 154-164.
5. Brandman, J., L. Sturm, J. White, and C. Williams, *A physical hash for preventing and detecting cyber-physical attacks in additive manufacturing systems.* Journal of Manufacturing Systems, 2020. **56**: p. 202-212.
6. Liu, C., C. Kan, and W. Tian. *An Online Side Channel Monitoring Approach for Cyber-Physical Attack Detection of Additive Manufacturing*. in *International Manufacturing Science and Engineering Conference*. 2020. American Society of Mechanical Engineers.
7. Zheng, Z., S. Xie, H. Dai, X. Chen, and H. Wang. *An overview of blockchain technology: Architecture, consensus, and future trends*. in *2017 IEEE international congress on big data (BigData congress)*. 2017. IEEE.
8. Bokhari, M.U. and Q.M. Shallal, *A review on symmetric key encryption techniques in cryptography.* International journal of computer applications, 2016. **147**(10).
9. Conti, M., N. Dragoni, and V. Lesyk, *A survey of man in the middle attacks.* IEEE communications surveys & tutorials, 2016. **18**(3): p. 2027-2051.
10. Shi, Z., C. Kan, W. Tian, and C. Liu, *A Blockchain-based G-code Protection Approach for Cyber-Physical Security in Additive Manufacturing.* Journal of Computing and Information Science in Engineering, 2021. **21**(4).
11. Zeltmann, S.E., N. Gupta, N.G. Tsoutsos, M. Maniatakos, J. Rajendran, and R. Karri, *Manufacturing and security challenges in 3D printing.* Jom, 2016. **68**(7): p. 1872-1881.

12. Chhetri, S.R. and M.A. Al Faruque, *Side channels of cyber-physical systems: Case study in additive manufacturing.* IEEE Design & Test, 2017. **34**(4): p. 18-25.
13. Villalobos, K., J. Suykens, and A. Illarramendi, *A flexible alarm prediction system for smart manufacturing scenarios following a forecaster–analyzer approach.* Journal of Intelligent Manufacturing, 2021. **32**(5): p. 1323-1344.
14. Wu, M., Z. Song, and Y.B. Moon, *Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods.* Journal of intelligent manufacturing, 2019. **30**(3): p. 1111-1123.
15. Shi, Z., A.A. Mamun, C. Kan, W. Tian, and C. Liu, *An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing.* Journal of Intelligent Manufacturing, 2022: p. 1-17.
16. Liu, C., Z. Kong, S. Babu, C. Joslin, and J. Ferguson, *An integrated manifold learning approach for high-dimensional data feature extractions and its applications to online process monitoring of additive manufacturing.* IISE Transactions, 2021. **53**(11): p. 1215-1230.
17. Liu, C., A.C.C. Law, D. Roberson, and Z.J. Kong, *Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication.* Journal of Manufacturing Systems, 2019. **51**: p. 75-86.
18. Dastoorian, R. and L.J. Wells, *A hybrid off-line/on-line quality control approach for real-time monitoring of high-density datasets.* Journal of Intelligent Manufacturing, 2021: p. 1-14.
19. Larsen, S. and P.A. Hooper, *Deep semi-supervised learning of dynamics for anomaly detection in laser powder bed fusion.* Journal of Intelligent Manufacturing, 2021: p. 1-15.
20. Ye, Z., C. Liu, W. Tian, and C. Kan, *In-situ Point Cloud Fusion for Layer-wise Monitoring of Additive Manufacturing.* Journal of Manufacturing Systems, 2021. **61**: p. 210-222.
21. Al Mamun, A., C. Liu, C. Kan, and W. Tian, *Securing cyber-physical additive manufacturing systems by in-situ process authentication using streamline video analysis.* Journal of Manufacturing Systems, 2022. **62**: p. 429-440.
22. Liu, C., W. Tian, and C. Kan, *When AI meets additive manufacturing: Challenges and emerging opportunities for human-centered products development.* Journal of Manufacturing Systems, 2022.
23. Li, Y., Z. Shi, and C. Liu, *Transformer-enabled Generative Adversarial Imputation Network with Selective Generation (SGT-GAIN) for Missing Region Imputation.* IISE Transactions, 2023(just-accepted): p. 1-19.
24. Li, R., M. Jin, and V.C. Paquit, *Geometrical defect detection for additive manufacturing with machine learning models.* Materials & Design, 2021. **206**: p. 109726.
25. Elhabashy, A.E., L.J. Wells, and J.A. Camelio, *Cyber-physical attack vulnerabilities in manufacturing quality control tools.* Quality Engineering, 2020. **32**(4): p. 676-692.
26. Elhabashy, A.E., L.J. Wells, J.A. Camelio, and W.H. Woodall, *A cyber-physical attack taxonomy for production systems: a quality control perspective.* Journal of Intelligent Manufacturing, 2019. **30**(6): p. 2489-2504.
27. Flank, S., A.R. Nassar, T.W. Simpson, N. Valentine, and E. Elburn, *Fast authentication of metal additive manufacturing.* 3D Printing and Additive Manufacturing, 2017. **4**(3): p. 143-148.
28. Komolafe, T., W. Tian, G.T. Purdy, M. Albakri, P. Tarazaga, and J. Camelio, *Repeatable part authentication using impedance based analysis for side-channel monitoring.* Journal of Manufacturing Systems, 2019. **51**: p. 42-51.
29. Wu, D., D.W. Rosen, L. Wang, and D. Schaefer, *Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation.* Computer-Aided Design, 2015. **59**: p. 1-14.
30. Yen, I.-L., S. Zhang, F. Bastani, and Y. Zhang. *A framework for IoT-based monitoring and diagnosis of manufacturing systems*. in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. 2017. IEEE.
31. Saeed, A., A. Ahmadinia, A. Javed, and H. Larijani, *Random neural network based intelligent intrusion detection for wireless sensor networks.* Procedia Computer Science, 2016. **80**: p. 2372-2376.
32. Zhang, Y., X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, *Blockchain-based trust mechanism for IoT-based smart manufacturing system.* IEEE Transactions on Computational Social Systems, 2019. **6**(6): p. 1386-1394.
33. Kurpjuweit, S., C.G. Schmidt, M. Klöckner, and S.M. Wagner, *Blockchain in additive manufacturing and its impact on supply chains.* Journal of Business Logistics, 2021. **42**(1): p. 46-70.
34. Aitzhan, N.Z. and D. Svetinovic, *Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams.* IEEE Transactions on Dependable and Secure Computing, 2016. **15**(5): p. 840-852.
35. Ghuli, P., U.P. Kumar, and R. Shettar, *A review on blockchain application for decentralized decision of ownership of IoT devices.* Advances in Computational Sciences and Technology, 2017. **10**(8): p. 2449-2456.
36. Bahga, A. and V.K. Madisetti, *Blockchain platform for industrial internet of things.* Journal of Software Engineering and Applications, 2016. **9**(10): p. 533-546.
37. Yu, C., L. Zhang, W. Zhao, and S. Zhang, *A blockchain-based service composition architecture in cloud manufacturing.* International Journal of Computer Integrated Manufacturing, 2020. **33**(7): p. 701-715.

38. Shafagh, H., L. Burkhalter, A. Hithnawi, and S. Duquennoy. *Towards blockchain-based auditable storage and sharing of iot data*. in *Proceedings of the 2017 on Cloud Computing Security Workshop*. 2017.
39. Schleich, B., N. Anwer, L. Mathieu, and S. Wartzack, *Shaping the digital twin for design and production engineering.* CIRP annals, 2017. **66**(1): p. 141-144.
40. Kennedy, Z.C., D.E. Stephenson, J.F. Christ, T.R. Pope, B.W. Arey, C.A. Barrett, and M.G. Warner, *Enhanced anti-counterfeiting measures for additive manufacturing: coupling lanthanide nanomaterial chemical signatures with blockchain technology.* Journal of Materials Chemistry C, 2017. **5**(37): p. 9570-9578.
41. Peterson, K., R. Deeduvanu, P. Kanjamala, and K. Boles. *A blockchain-based approach to health information exchange networks*. in *Proc. NIST Workshop Blockchain Healthcare*. 2016.
42. Merkle, R.C. *One way hash functions and DES*. in *Conference on the Theory and Application of Cryptology*. 1989. Springer.
43. Dasgupta, D., J.M. Shrein, and K.D. Gupta, *A survey of blockchain from security perspective.* Journal of Banking and Financial Technology, 2019. **3**(1): p. 1-17.
44. Gaubatz, G., J.-P. Kaps, and B. Sunar. *Public key cryptography in sensor networks—revisited*. in *European Workshop on Security in Ad-Hoc and Sensor Networks*. 2004. Springer.
45. Kerry, C.F. and P.D. Gallagher, *Digital signature standard (DSS).* FIPS PUB, 2013: p. 186-4.
46. Rivest, R.L., A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, 1978. **21**(2): p. 120-126.
47. Koblitz, N., A. Menezes, and S. Vanstone, *The state of elliptic curve cryptography.* Designs, codes and cryptography, 2000. **19**: p. 173-193.
48. Mahto, D. and D.K. Yadav, *Performance Analysis of RSA and Elliptic Curve Cryptography.* Int. J. Netw. Secur., 2018. **20**(4): p. 625-635.
49. Saho, N.J.G. and E.C. Ezin. *Comparative study on the performance of elliptic curve cryptography algorithms with cryptography through RSA algorithm*. in *CARI 2020-Colloque Africain sur la Recherche en Informatique et en Mathématiques Apliquées*. 2020.
50. Cheng, C., R. Lu, A. Petzoldt, and T. Takagi, *Securing the Internet of Things in a quantum world.* IEEE Communications Magazine, 2017. **55**(2): p. 116-120.
51. Collberg, C., C. Thomborson, and D. Low, *A taxonomy of obfuscating transformations*. 1997, Department of Computer Science, The University of Auckland, New Zealand.
52. Bakken, D.E., R. Rarameswaran, D.M. Blough, A.A. Franz, and T.J. Palmer, *Data obfuscation: Anonymity and desensitization of usable data sets.* IEEE Security & Privacy, 2004. **2**(6): p. 34-41.
53. Gomatam, S. and A. Karr, *Distortion measures for categorical data swapping.* J. Official Statist, 2003.
54. Boneh, D. and H. Shacham, *Fast variants of RSA.* CryptoBytes, 2002. **5**(1): p. 1-9.