IISE TRANSACTIONS
https://doi.org/10.1080/24725854.2023.2291728

Taylor & Francis
Taylor & Francis Group

Distributed cryptosystem for service-oriented smart manufacturing

Alexander Krall[a], Daniel Finke[b], and Hui Yang[a]

[a]Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA, USA; [b]Applied Research Lab, The Pennsylvania State University, University Park, PA, USA

ABSTRACT
Advanced sensing and cloud systems propel the rapid advancements of service-oriented smart manufacturing. As a result, there is widespread generation and proliferation of data in the interest of manufacturing analytics. The sheer amount and velocity of data have also attracted a myriad of malicious parties, unfortunately resulting in an elevated prevalence of cyber-attacks whose impacts are only gaining in severity. Therefore, this article presents a new distributed cryptosystem for analytical computing on encrypted data in the manufacturing environment, with a case study on manufacturing resource planning. This framework harmonizes Paillier cryptography with the Alternating Direction Method of Multipliers (ADMM) for decentralized computation on encrypted data. Security analysis shows that the proposed Paillier-ADMM system is resistant to attacks from external threats, as well as privacy breaches from trusted-but-curious third parties. Experimental results show that smart allocation is more cost-effective than the benchmarked deterministic and stochastic policies. The proposed distributed cryptosystem shows strong potential to leverage the distributed data for manufacturing intelligence, while reducing the risk of data breaches.

ARTICLE HISTORY
Received 31 May 2023
Accepted 29 November 2023

KEYWORDS
Manufacturing planning; Paillier cryptosystem; data analytics; cyber manufacturing

1. Introduction

Digital integration has become pervasive in the manufacturing space as industry giants push deeper into (and even past) Industry 4.0. The principal driver in achieving this end entails the enhanced connectivity among sensing devices, institutions, and decision-makers. In a highly integrated smart environment, factory "things" (e.g., machines, material handling systems, inventory) will proliferate data to assist in the generation of artificial intelligence and digital twin (Lee and Yang, 2023). Distributed data and information sharing, unfortunately, causes security and privacy concerns, especially when service communications occur among institutions (Lee *et al.*, 2023). A report from Barracuda Networks, as of 2022, shows that the state of industrial security is dire: 94.38% of manufacturers indicated that their operations were afflicted by a cyber-incident within the previous 12 months (Bourne, 2022). These incidents have significant impacts on industrial organizations, with 87% of those surveyed reporting operational disruption lasting for a day or more. Of these disruptions, 34% of organizations experienced a compromised supply chain. Likewise, 31% of organizations have experienced data theft.

Smart manufacturing increasingly relies on distributed computing techniques to handle large datasets in support of service architectures. Nevertheless, an attacker eavesdropping on unsecured communication channels can ascertain information pertinent to both machines and production processes. Theft of industrial information can be leveraged by malicious organizations to conduct attacks on supply-chain systems or gain competitive advantages. Even in the face of these risk factors, institutions pursue data-driven coordination to mitigate costs and remain viable economically. After all, participating in the open market necessitates adaptation to technological drivers of optimized production and distribution.

Therefore, there is an urgent need to manage the benefits of analytical insights and the potential risks of data breaches to manufacturing services. Cryptography practices are employed to secure data in transit, which reduces the risk of man-in-the-middle attacks. Despite the utility of cryptography, encrypting and decrypting data necessitates computation expenditure. Elevated frequency of encryption and decryption translates into greater operational overhead. Furthermore, one institution must trust all relevant parties in their network to securely handle all decrypted data because a system is only as secure as its weakest link. Third-party organizations, even if not outright malicious, may even try to learn as much pertinent information as they can about their peers to boost their own competitive viability. To address these security and privacy concerns, homomorphic encryption techniques were developed. Homomorphic encryption allows mathematical operations to be performed on encrypted data without the need for intermediary decryption. Under this paradigm, data can be secured both in transit and at the destination.

Nonetheless, little has been done to harmonize homomorphic encryption algorithms with distributed computation

CONTACT Hui Yang huy25@psu.edu

Supplemental data for this article can be accessed online at https://doi.org/10.1080/24725854.2023.2291728.

to meet the needs of a service-oriented smart manufacturing. Therefore, this article presents a new distributed cryptosystem for analytical computing on encrypted data in the manufacturing environment, with a case study on manufacturing resource planning. Specifically, our core contributions are as follows:

1. ***Service-oriented manufacturing planning.*** The act of supplying a customer may be thought of as a service. Products flow through a supply chain in a way that is analogous to electricity distribution. Transmission operations pass materials from one party to another. Propagation operations cause the material to come into being. Within a supply-chain system, a network of suppliers is responsible for supplying a network of factories. Materials, upon reaching the factory, are then allocated to products. In light of this service-oriented view of the supply chain, there is an opportunity to minimize the overall system cost through coordination between the supplier network and factories. Thus, we develop a new distributed optimization formulation on encrypted data to drive manufacturing resource planning in a network of suppliers and smart factories.
2. ***Distributed cryptosystem for smart manufacturing.*** Decentralizing the computing tasks (or edge/fog computing) is conducive to improving efficiency and robustness of decision-making by hastening the availability of analytical insights. To this end, distributed optimization within the supply chain necessitates multiparty information coordination. Nonetheless, sharing information gives way to privacy and security risks. As a result, we develop a distributed cryptosystem that harmonizes homomorphic encryption techniques with the Alternating Direction Method of Multipliers (ADMM) for smart manufacturing applications. Case study and experimental results show that the proposed cryptosystem is more cost-effective to improve the smartness of manufacturing resource planning than the benchmarked deterministic and stochastic policies.

The development of a distributed and service-oriented cryptosystem for smart manufacturing will attenuate production costs, preserve industrial privacy, and maintain the computational efficiency.

2. Research background

2.1. Manufacturing planning and control

Historically, push and pull production control systems are presented as two disparate drivers of manufacturing praxis (Yang, 2013). The key difference between these paradigms is how work releases are triggered. Push systems schedule work releases based on customer demand, whereas pull systems authorize work releases based on system status (Khojasteh, 2016).

Material Requirements Planning (MRP) is a well-known implementation of a push system, and is considered most useful when final products have intermediate dependencies. The primary function of MRP is to translate a master production schedule, bill of materials, and inventory data into a schedule of planned order releases for all end products and intermediaries. According to Hopp and Spearman (2008) MRP systems are prone to capacity infeasibility issues because the overall model assumes infinite capacity. Thus, capacity feasibility must be ensured in the master production schedule. Furthermore, there tends to be pressure to increase the planned lead times, due to exorbitant penalties when jobs are late. Lengthening lead times ultimately inflates Work-In-Progress (WIP) inventories (Lee and Yang, 2023). Additionally, the MRP model is inherently volatile to small changes in the master production schedule. In a very counterintuitive fashion, decreases in demand can result in infeasibilities, even when MRP output was previously feasible.

To address these issues, the MRP II workflow depicted in Figure 1 was developed. Under MRP II the master production schedule is dependent on an aggregate production plan, which determines the production quantity and timing of product families. Planning for product families rather than individual products mitigates demand variance. Once generated, the aggregate plan is broken down through a disaggregation process to produce the master production schedule for individual products. The MRP II framework has an internal MRP I module, which takes the master production schedule as input. Lot sizes and due date plans obtained from the MRP I module are leveraged to govern shop floor-level production control. Eventually, institutions sought a more integrated approach to handle business planning to incorporate modern supply chain management practices. Thus, MRP II gave way to Enterprise Resource Planning (ERP), which enables manufacturing factories to control all business operations digitally (i.e., manufacturing, distribution, accounting, financial, and personnel).

Despite their level of sophistication, push systems have fundamental issues underlying their basic models (Ye *et al.*, 2021). Even some of the best ERP systems assume infinite capacity and fixed lead times. As a result, pull systems were developed to address the shortcomings of modern push systems. A pull system's primary concern is the minimization of WIP inventory. To this end, a new order initiates production. In a Kanban pull system, inventory is limited to a fixed maximum at each workstation, which represents a single process and its output buffer. When a part is pulled out of any output buffer, the preceding workstation is given an authorization signal to begin production. During the production process, this workstation will consume parts from upstream inventory buffers. In doing so, the workstation relays production authorizations to upstream workstations and drives the replacement of consumed inventories. This process initiates an authorization cascade that propagates to all workstations in the system. Variations of Kanban include base-stock and constant WIP systems. Overall, pull systems offer multifaceted benefits including, but not limited to, reduced inventory, shorter lead times, less variability, shorter time to detect quality issues, greater flexibility (Yang *et al.*, 2019).

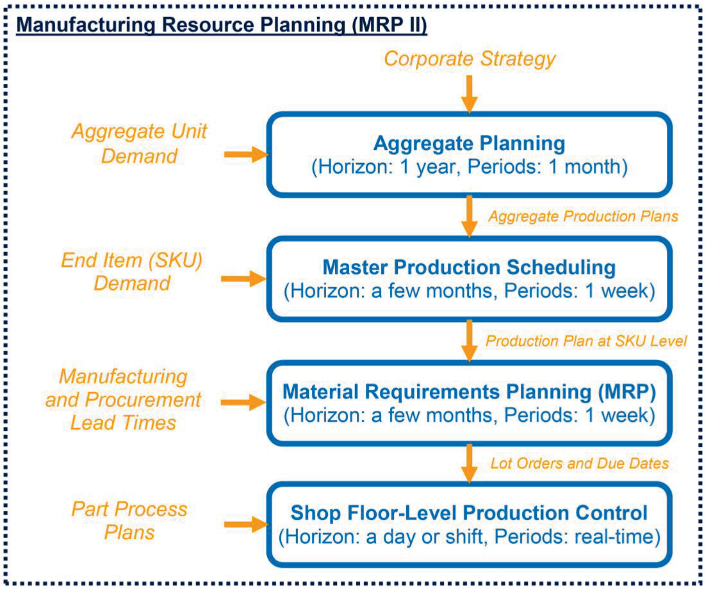
Manufacturing Resource Planning (MRP II)
Corporate Strategy
Aggregate Unit Demand
Aggregate Planning
(Horizon: 1 year, Periods: 1 month)
Aggregate Production Plans
End Item (SKU) Demand
Master Production Scheduling
(Horizon: a few months, Periods: 1 week)
Production Plan at SKU Level
Manufacturing and Procurement Lead Times
Material Requirements Planning (MRP)
(Horizon: a few months, Periods: 1 week)
Lot Orders and Due Dates
Part Process Plans
Shop Floor-Level Production Control
(Horizon: a day or shift, Periods: real-time)

Figure 1. Production planning workflow under MRP II.

Technological innovations have opened opportunities for more cooperative planning in manufacturing, which result in the hybridization of push and pull strategies. To this end, the rise of cloud computing has given way to cloud manufacturing (also known as manufacturing-as-a-service) (Yang *et al.*, 2021). Balta *et al.* (2018) developed a cloud-based platform for connecting user production requests to manufactures. Under this framework, manufacturing steps are abstracted into service requests. Furthermore, according to Liu *et al.* (2019) manufacturing services are more diverse, having longer execution and greater variety in delivery when compared with general computing services. Additionally, the common understanding of key factors pertinent to cloud manufacturing is limited, including the concept, operation model, service mode, technology system, architecture, and essential characteristics. Yang *et al.* (2019) present a cyber-physical framework that integrates cloud computing with the internet of things to develop a virtual machine network for manufacturing process monitoring and control.

In spite of recent advances, the following gaps remain in the state of the art. Enterprises in the supply chain conventionally schedule (or authorize) production in isolation from each other. Push systems attempt to be proactive, but are plagued by limited visibility into downstream requirements. These push systems often have excessive inventories as a result. On the other hand, pull systems, are entirely demand-reactive. A manufacturer may experience backlogged demand if production capacity is insufficient. Backlog may be triggered by supply shocks, demand shocks, and large orders. Technological innovations provide a higher level of supply chain coordination that allows for push–pull hybridization. However, institutions always aim to optimize their own costs without necessarily considering the implications for suppliers. Therefore, there is an urgent need to develop a more collaborative and secure architecture for smart manufacturing. Very little has been done to develop a distributed cryptosystem for manufacturing resource planning. In addition, manufacturing is often seen in juxtaposition to the service industry. The prevailing view is that products are tangible, whereas services are intangible. Nonetheless, this rigid dichotomy is limiting, as insights from the service industry tend to benefit manufacturing. Furthermore, service-oriented manufacturing is not well-defined in the state of the art. Thus, there is an opportunity to integrate service-oriented planning with manufacturing control.

2.2. Distributed computation

In the absence of parallel processing techniques, optimization problems are solved in series, which introduces significant computational overhead (Ye *et al.*, 2023). Therefore, a map-reduce framework is introduced to split a large dataset into subgroups. These segments are then distributed or "mapped" to processing units for computation. After computation has finished, the "reduce" step collates the outputs. Furthermore, Kan *et al.* (2023) design a large-scale machine processing method for the Industrial Internet of Things (IIoT) (Kan *et al.*, 2018). A dynamic warping algorithm is leveraged to determine the dissimilarity of machine signatures. Based on these dissimilarity measures, a stochastic network embedding algorithm is designed to construct a large-scale network of IIoT machines.

Further, the ADMM was developed to partition convex optimization problems into smaller subproblems that are easier to solve (Boyd *et al.*, 2011). ADMM allows optimization to be parallelized when variables (or constraints) are separable. Likewise, Kranning *et al.* (2014) leverage the concept of proximal message passing to handle dynamic optimal power flow problems in a decentralized setting. Their method boasts quick execution and large scalability, and is shown to converge to a solution when the device objective and constraints are both convex. Also, Paillier-augmented ADMM formulations were proposed for power systems

(Errapotu *et al*., 2018; Zhang *et al*., 2018; Wu *et al*., 2021; Huo and Liu, 2022; Si *et al*., 2023).

However, a gap remains in the integration of distributed encryption and computing with manufacturing resource planning that allows for push–pull hybridization. Note that manufacturing is vastly different from power systems. Formulating ADMM greatly depends on the domain-specific optimization models. In the domain of smart manufacturing, there are high degrees of coupling between decision variables, constraints, and cost functions, due to the system architecture. Thus, discrete-part manufacturing is different from continuous-flow power systems. In addition, distributing data subsets across many machines introduces significant transmission overhead. Computing methods that entail the distribution of subproblems to processing units may suffer from poor convergence rates. It is not uncommon that data transfer often assumes a completely trusted environment. In reality, an environment is subject to privacy and security vulnerabilities, which call upon the mitigation of malicious exfiltration.

2.3. Cybersecurity

In light of the modern landscape of cyber threat, NIST designed a framework to address cybersecurity risks in manufacturing environments (Stouffer *et al*., 2020). The framework embodies five functions: identify, protect, detect, respond, recover. Each of these functions has various categories and subcategories (e.g., "identify" encapsulates asset management). An actionable implementation of each subcategory is realized through a manufacturing profile, which further considers five business objectives: maintaining environmental safety, human safety, production goals, quality of product, and sensitive information. Potential impacts to the subcategories are scored as low, moderate, or high. Despite its utility, this profile relies on a manufacturer's internal processes to drive cybersecurity decision-making and prioritization. Biehler *et al.* (2024) also designed a risk assessment framework for cyber-physical systems known as Stealthy Attack Generation (SAGE). The SAGE system models covert attacks whose goal is to maximize damage, avoid detection, and minimize attack cost.

Within the scope of information transfer, Miller *et al.* (2017) designed a Minimum Information Model (MIM) for 3D annotated product definitions. Within model-based enterprises, it is necessary to share information to carry forth a given workflow. Minimizing shared information ensures that any generated data can be easily verified, analyzed, and interpreted. Although the MIM does not explicitly have security in mind, the minimization of information sharing can have the implicit effect of mitigating the leakage of sensitive information in accordance with NIST principles.

In addition to NIST, SAGE, and MIM-derived insights, a variety of security innovations have been developed. Vedeshin *et al.* (2020) devised a replacement for asymmetric encryption for personal manufacturing. Their technique utilizes an unkeyed cryptographic hash function that is collision, second pre-image, and pre-image resistant. This hash function does not have an inverse function and allows for secure distributed file storage and transfer. Li *et al.* (2021) developed a degradation process to detect covert attacks against supervisory control and data acquisition (SCADA) systems. Krall *et al* (2021) designed Mosaic Gradient Perturbation (MGP) to preserve the privacy of predictive models. The MGP technique upholds differential privacy standards while protecting sensitive data attributes against model inversion attacks. The model can be fine-tuned to manage trade-offs between model performance and attack accuracy.

Homomorphic encryption schemes allow for computations to be performed on encrypted data without the need for decryption. There are a variety of properties a homomorphic encryption scheme possesses:

1. **Correct decryption:** Errorless decryption of a ciphertext into the correct plaintext.
2. **Correct evaluation:** With overwhelming probability, the decryption of a homomorphic evaluation of a permitted circuit produces the correct result.
3. **Compactness:** The size of ciphertexts does not grow substantially through successive homomorphic operations.

Partially homomorphic encryption schemes only require the correct decryption and correct evaluation properties. Fully homomorphic encryption requires compactness, correctness (of decryption and evaluation), and support for arbitrary circuits (Armknecht *et al*., 2015). Paillier *et al.* (1999) created a partially homomorphic encryption scheme that allows for the addition of two encrypted inputs without decryption Furthermore, Li *et al.* (2018) developed an efficient fully homomorphic encryption algorithm that optimizes the decryption process for a large plaintext space.

Despite the NIST and MIM frameworks being geared towards preventing the leakage of sensitive data, maximal amounts of information are leveraged to provide useful analytical insights. In the traditional encryption framework, remote data subset processing (in the absence of homomorphic cryptography) requires information to be decrypted before it can be used. Should a processing unit be compromised by a malicious agent, the data is liable to be exfiltrated. Also, the elevated frequency of encrypting and decrypting, resulting from computational needs, introduces significant computational overhead to a system and results in reduced expediency. Finally, due to the limited number of available mathematical operations, homomorphic encryption schemes either require algorithmic compatibility or necessitate significant workflow modifications.

3. Research methodology

3.1. System architecture

This article focuses on a smart manufacturing system whose objective is to minimize the expenditure of smart factories and their material suppliers for make-to-order products.

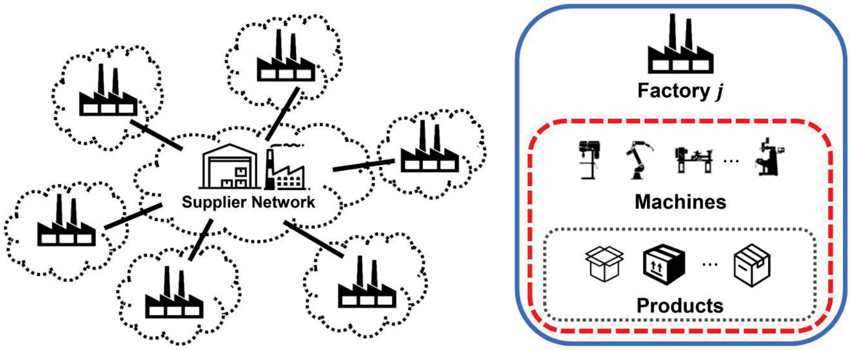

Figure 2. The architecture for manufacturing planning and control. Within a smart factory, materials are allocated according to the demand of products or product families, then assigned to machines.

Under this framework, each factory will need to update its material allocation information and relay these updates to suppliers, which is often sensitive and proprietary. As shown in Figure 2, a series of factories equipped with smart machines are connected to a greater supplier network. Data collected by these smart machines give insight into material allocations assigned to different products or product families. The allocation architecture, therefore, has additional complexity and vulnerabilities for privacy leaks.

Suppose we have a set of smart factories in a community, $\mathcal{J}$ indexed by $j = 1, ..., J$. Each factory delivers a set of products $\mathcal{K}$ indexed by $k = 1, ..., K$, and will have multiple suppliers and require different types of materials. We consider a supplier network $\mathcal{S}$ that seeks to minimize its production cost of a single material. It is common that each product family is supported with a specific supplier network (Yang *et al.*, 2021). For multiple product families, there will be multiple material plans, each of which is, however, associated with a supplier network. As such, distributed cryptosystems could be established for each network of suppliers. Although multiple networks of suppliers could be entangled together, the risk of privacy breach will likely increase. From the perspective of a manufacturing factory, the material planning focuses more on a specific supplier network when dealing with the customer demand. In other words, a factory is often in a smaller scale than the enterprise and is rarely handling multiple supplier networks at the same time. If multiple supplier networks are mixed for decision-making, there is a need to scale beyond the factory level and formulate new case studies that are worthy of further investigations. The design of a system architecture necessitates a trusted third-party key generator who is responsible for assigning public–private key pairs to suppliers. In this framework, suppliers only have access to their public key. Private keys are maintained by the key generator. Suppliers will distribute their public key to customers who seek to opt into the proposed system. Furthermore, let $x_{j,k}$ represent the smart factory j's material demand information for product k. The consumption information for all of factory j's products (x_j) is optimized with the assistance of the supplier network.

The objective function of the factory for time period t is formulated based on the material scheduling of smart machines and is given by $U_j^t(\cdot)$. This objective function is designed to be non-decreasing, non-negative, and convex in x_j. An example of the utility function is given to be

$$U_j^t(x_j) = \sum_{k\in\mathcal{K}} \frac{x_{j,k}}{\beta_{j,k}^t} \tag{1}$$

where $\beta_{j,k}^t$ is the backorder cost of product k within factory j at time t. In general, the function penalizes inadequate allocation of materials to products with a high backorder cost. Other convex, non-negative, and non-decreasing functions representing the material allocation elasticity can be utilized.

The objective function of the supplier network for time period t is based on variations in production and distribution costs and is given by $C_k^t(\cdot)$. This objective function is non-decreasing, non-negative and convex in x_k. An example of the cost function is

$$C_k^t(x_k) = \sum_{j\in\mathcal{J}} \left(\Lambda_j^t + c^t\right) \cdot x_{j,k} \tag{2}$$

where c^t is the supplier network's material production cost and at time t and Λ_j^t is the distribution cost to the factory j at time t.

3.2. Manufacturing planning and control

Service-oriented governance of the smart manufacturing system is handled through the minimization of the supplier network and factory user's costs by solving the following optimization problem at a particular time period t, which will encapsulate a given planning horizon:

$$\min_{\{x_j\}_{j=1}^J} \sum_{k\in\mathcal{K}} C_k(x_k) + \alpha \sum_{j\in\mathcal{J}} U_j(x_j) \tag{3}$$

subject to

$$\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} x_{j,k} \le \Omega \tag{4}$$

$$x_{j,k}^{\min} \le x_{j,k} \le x_{j,k}^{\max}, \ \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{5}$$

$$\textstyle\sum_{k\in\mathcal{K}} x_{j,k} = \Theta_j, \ \forall j \in \mathcal{J} \tag{6}$$

$$x_{j,k} \ge 0, \ \forall\ j \in \mathcal{J}, k \in \mathcal{K} \tag{7}$$

where $\alpha > 0$ controls the influence of the factory costs during optimization, Ω is the material production capacity of the supplier network, and Θ_j is factory j's order quantity. The coupling of $x_{j,k}$ between downstream smart factories prevents the problem from being solved in a distributed fashion. Hence, we seek to address this limitation through an equivalent separable ADMM representation.

3.3. ADMM

We propose to solve the convex optimization problem – service-oriented manufacturing planning – with ADMM by breaking it into subproblems. These subproblems require less computational resources to solve than the whole. The generic optimization problem we consider follows the following form:

$$\min_{x,z} f(x) + g(z) \tag{8}$$

subject to

$$Ax + Bz = c \tag{9}$$

where $x \in \mathbb{R}^n, z \in \mathbb{R}^m, c \in \mathbb{R}^p$ and the matrices $A \in \mathbb{R}^{p\times n}, B \in \mathbb{R}^{p\times m}$. Functions f and g are convex, close, and proper. The scaled augmented Lagrangian is expressed as:

$$\mathcal{L}_\varrho(x,z,\mu) = f(x) + g(z) + \frac{\varrho}{2}\|Ax + Bz - c + \mu\|_2^2 \tag{10}$$

Note that $\varrho > 0$ is the penalty parameter and μ is the scaled dual variable. The variables x and z are updated, in order, according to:

$$x^{i+1} = \underset{x}{\operatorname{argmin}} f(x) + \frac{\varrho}{2}\|Ax + Bz^i - c + \mu^i\|_2^2 \tag{11}$$

$$z^{i+1} = \underset{z}{\operatorname{argmin}} g(z) + \frac{\varrho}{2}\|Ax^{i+1} + Bz - c + \mu^i\|_2^2 \tag{12}$$

$$\mu^{i+1} = \mu^i + Ax^{i+1} + Bz^{i+1} - c \tag{13}$$

The separability of the objective function in terms of its variables x and z provides ADMM with an advantage. Dual variable updates typically necessitate solving a regularization function for both x and z at the same time. ADMM solves this issue by first solving for x with a fixed z. Next, ADMM solves for z with a fixed x. The algorithm then proceeds to update the dual variable before repeating the entire process. Although the method does not provide an exact minimal solution, it does converge to an optimal value under particular assumptions.

3.4. Distributed ADMM for smart manufacturing

To facilitate solving the optimization problem in a distributed manner, we introduce auxiliary variables for the factory's machines (z_k) and enforce $z_{j,k} = x_{j,k}$. The formulation takes the following form:

$$\min_{\{x_j\}_{j=1}^J} \sum_{k\in\mathcal{K}} C_k(z_k) + \alpha \sum_{j\in\mathcal{J}} U_j(x_j) \tag{14}$$

subject to

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} x_{j,k} \le \Omega \tag{15}$$

$$x_{j,k}^{\min} \le x_{j,k} \le x_{j,k}^{\max}, \;\; \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{16}$$

$$\textstyle\sum_{k\in\mathcal{K}} x_{j,k} = \Theta_j, \;\; \forall j \in \mathcal{J} \tag{17}$$

$$x_{j,k} = z_{j,k}, \;\; \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{18}$$

$$x_{j,k}, z_{j,k} \ge 0, \;\; \forall \; j \in \mathcal{J}, k \in \mathcal{K} \tag{19}$$

Because the optimization problem is in the ADMM form, it can be solved in a distributed manner. The decision variables x_j and z_k are arranged into two groups, corresponding to the factory user and supplier network updates, respectively. Variables in each group are optimized in parallel. Each user solves for x_j and the supplier network solves for z_k. We now define sets:

$$\mathcal{A}_j = \left\{ x_j \;\middle|\; \begin{array}{l} x_{j,k}^{\min} \le x_{j,k} \le x_{j,k}^{\max} \\ \textstyle\sum_{k\in\mathcal{K}} x_{j,k} = \Theta_j; \\ x_{j,k} \ge 0, \;\; \forall k \in K \end{array} \right\}$$

$$\mathcal{B}_k = \{ z_k | z_{j,k} \ge 0, \;\; \forall j \in \mathcal{J} \}.$$

Thus, we have $\mathcal{A} = \cup_{j=1}^J \mathcal{A}_\mathrm{j}$ and $\mathcal{B} = \cup_{k=1}^K \mathcal{B}_\mathrm{k}$. We first calculate the partial Lagrangian, which induces Lagrange multipliers for the auxiliary constraint given by (18):

$$\mathcal{L}_\varrho\left(\{x_j\}_{j=1}^J, \{z_k\}_{k=1}^K, \{\mu_{j,k}\}_{j=1,k=1}^{J,K}\right) = \sum_{k\in\mathcal{K}} C_k(z_k) + \alpha \sum_{j\in\mathcal{J}} U_j(x_j) + \frac{\varrho}{2}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} \|x_{j,k} - z_{j,k} + \mu_{j,k}\|_2^2 \tag{20}$$

where $\mu_{j,k}$ is the scaled Lagrangian multiplier. The decision variables x_j and z_k are arranged into two groups and updated in an iterative fashion, as seen in Algorithm 1.

Algorithm 1 Distributed ADMM for Manufacturing Planning

```
Input: $\{C_k(\cdot)\}_{k=1}^K, \{U_j(\cdot)\}_{j=1}^J, \Omega,$
  $\{x_{j,k}^{\min}\}_{j,k=1}^{J,K}, \{x_{j,k}^{\max}\}_{j,k=1}^{J,K}$
  $i = 0$
Output: $\{x_j\}_{j=1}^J, \{z_k\}_{k=1}^K$
Initialize $\{x_j\}_{j=1}^J, \{z_k\}_{k=1}^K, \{\mu_{j,k}\}_{j=1,k=1}^{J,K} = 0$
Set $i = 1$
while not converging do
    with the factories
        for $j = 1, \ldots, J$ do in parallel for factory owners
            $\min_{x_j \in \mathcal{A}_j} \frac{\varrho}{2} \sum_{k\in\mathcal{K}} \|x_{j,k} - z_{j,k}^i + \mu_{j,k}^i\|_2^2 + \alpha U_j(x_j)$
            Send $x_{j,k}^{i+1}$ to the supplier network
        end for
    end with
    with the supplier network
        for $k = 1, \ldots, K$ do in parallel at supplier network
            $\min_{z_k \in \mathcal{B}_k} C_k(z_k) + \frac{\varrho}{2}\sum_{j\in\mathcal{J}} \|x_{j,k}^{i+1} - z_{j,k}^i + \mu_{j,k}^i\|_2^2$
        end for
        Obtain optimal solution $x_{j,k}^{i+1}, z_{j,k}^{i+1}$
        $\mu_{j,k}^{i+1} = \mu_{j,k}^i + x_{j,k}^{i+1} - z_{j,k}^{i+1}$
        Broadcast optimal solution to all factory owners
    end with
    Adjust penalty parameter $\varrho$ if necessary
    Set $i = i + 1$
end while
return $\{x_j\}_{j=1}^J, \{z_k\}_{k=1}^K$
```

3.4.1. Smart factory updates

At iteration i, x_j is updated by solving:

$$\min_{\{x_j\}_{j=1}^J \in \mathcal{A}} \frac{\varrho}{2}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} \|x_{j,k} - z_{j,k}^i + \mu_{j,k}^i\|_2^2 + \alpha\sum_{j\in\mathcal{J}} U_j(x_j) \tag{21}$$

This step is handled at independent computational units locally. Each computing unit j solves a stochastic optimization problem as follows:

$$\min_{x_j \in \mathcal{A}_j} \frac{\varrho}{2}\sum_{k\in\mathcal{K}} \|x_{j,k} - z_{j,k}^i + \mu_{j,k}^i\|_2^2 + \alpha U_j(x_j) \tag{22}$$

3.4.2. Supplier network

At iteration i, z_k is updated by solving:

$$\min_{\{z_k\}_{k=1}^K \in \mathcal{B}} \sum_{k\in\mathcal{K}} C_k(z_k) + \frac{\varrho}{2}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} \|x_{j,k}^{i+1} - z_{j,k} + \mu_{j,k}^i\|_2^2 \tag{23}$$

The supplier network will determine each z_k independently by solving the following in parallel:

$$\min_{z_k \in \mathcal{B}_k} C_k(z_k) + \frac{\varrho}{2}\sum_{j\in\mathcal{J}} \|x_{j,k}^{i+1} - z_{j,k} + \mu_{j,k}^i\|_2^2 \tag{24}$$

Lastly, dual variables are updated by the supplier network as follows:

$$\mu_{j,k}^{i+1} = \mu_{j,k}^i + x_{j,k}^{i+1} - z_{j,k}^{i+1} \tag{25}$$

3.5. Threat model

Solving the smart manufacturing problem with ADMM results in the collection and transmission of large amounts of material allocation data. As depicted in Figure 3, an attacker may be able to initiate a man-in-the-middle attack by eavesdropping on the connection between two processing units. We consider make-to-order products in our system architecture. Thus, material allocation information pertinent to these products reveal information about sensitive industrial processes, such as machine usage and interaction patterns. Data exfiltration in this industrial context not only includes corporate espionage, but may aid an adversary in levying successful attacks on critical digital infrastructure. For example, an attacker can impersonate an industrial device and transmit inaccurate allocation data to factory administrators. If done on a massive scale, an adversary may cause significant disruptions to national and global supply chains.

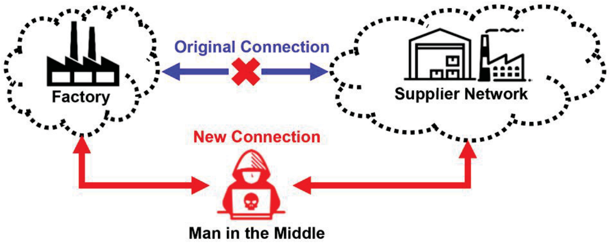

Figure 3. Smart manufacturing threat model showing a man-in-the-middle attack.

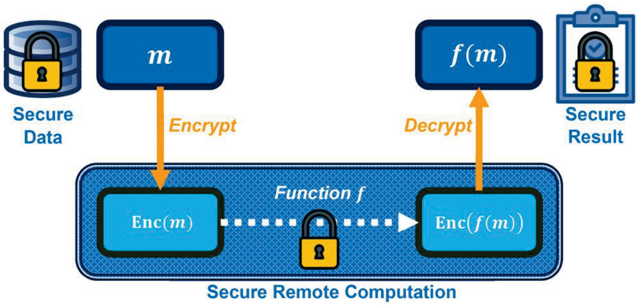

Figure 4. Encryption and decryption using Paillier cryptography.

When sensitive allocation information is passed between institutions within the supply chain, an attacker may attempt to add design features (e.g., extra steps to a process to lower the integrity of a part). Upon deciphering inter-institution communication, an attacker may also be able to piece together the intimate structure of the entire supply chain. Thus, surveillance aids an adversary in choosing targets as well as attack payloads such that the maximum amount of damage or disruption may be achieved.

In this article we consider an adversary that is actively trying to learn factory material allocation data. Additionally, we assume that both the supplier network and key generator are trustworthy, but curious. Thus, the supplier network will solve the optimization problem honestly, but will attempt to extrapolate product allocation data. Therefore, there is an urgent need to secure factory data from both external adversaries and the supplier network.

3.6. Paillier cryptosystem

Homomorphic encryption offers the opportunity to perform computations on encrypted data without the need for decryption as shown in Figure 4. To this end, we leverage the Paillier cryptosystem to encrypt manufacturing data for service-oriented optimization, which satisfies the indistinguishable and additive properties:

1. *Indistinguishability*. The encryption function, $E(x)$, is calculated with a random number, r. Therefore, should

the same plaintext be encrypted on two separate occasions, the two resultant ciphertexts will appear different. Because each occasion would utilize a different random number, we would be unable to tell if original plaintexts are the same without first decrypting them.

2. *Additive homomorphism*. The encryption function, $E(\cdot)$, is additively homomorphic if $E(x_1 + x_2) = E(x_1) + E(x_2)$.

We now detail the Pallier workflow. Let the public key be given by $\kappa = \langle n, \omega \rangle$, with a corresponding private key, $\sigma = \langle \lambda, \zeta \rangle$. Encryption can be performed by factories with the public key. However, decryption can only be performed by the carrier of the private key. The Pallier cryptosystem depends on two initial parameters b and q. Ciphertext $[\![x]\!]$ of plaintext x is produced through the encryption process:

$$[\![x]\!] = E(x) = \omega^x r^n \ \textbf{mod} \ n^2 \tag{26}$$

where $r \in \mathbb{Z}_n^*$, $\omega \in \mathbb{Z}_{n^2}^*$ are random integers coprime to n and n^2, respectively, with $n = bq$. The decryption process is carried forth with the following:

$$x = D([\![x]\!]) = \left[\zeta \cdot H\left([\![x]\!]^\lambda \ \textbf{mod} \ n^2\right)\right] \ \textbf{mod} \ n \tag{27}$$

where $\lambda = LCM(b-1, q-1)$, $\zeta = [H(\omega^\lambda \ \textbf{mod} \ n^2)]^{-1} \textbf{mod} \ n$, and $H(y) = \left\lfloor \frac{y-1}{n} \right\rfloor$. Note that $\lfloor \cdot \rfloor$ is the floor function.

Theorem 1: *The Pallier cryptosystem has the additive homomorphism property.*

Proof We first take the product of the ciphertexts:

$$\begin{aligned}[\![x_1]\!] \cdot [\![x_2]\!] &= (\omega^{x_1} r^n \ \textbf{mod} \ n^2)(\omega^{x_2} r^n \ \textbf{mod} \ n^2) \\ &= \omega^{x_1 + x_2} r^n \ \textbf{mod} \ n^2\end{aligned}$$

Thus, decrypting the result would yield the sum $x_1 + x_2$. □

Additional useful homomorphic properties of Paillier cryptography include:

- Adding a plaintext x_2 to a ciphertext $[\![x_1]\!]$ with $D([\![x_1]\!] \cdot \omega^{x_2}) = x_1 + x_2$.
- Multiplying a ciphertext $[\![x_1]\!]$ by a plaintext x_2 with $D([\![x_1]\!]^{x_2}) = x_1 \cdot x_2$.
- Refreshing the value of ciphertext $[\![x]\!]$ with $[\![x]\!] + [\![0]\!]$.

3.7. Distributed ADMM computing on encrypted data

We now harmonize Algorithm 1 (distributed and service-oriented smart manufacturing planning) with Paillier cryptography to mitigate the existent threat model. Initially, the key generator will generate the supplier's public key (κ_S) and private key (σ_S) pair. Each factory will receive the public key (κ_S) of each supplier within the network, which will be used for encrypting purposes. Should the supplier network seek to solve the optimization problem of manufacturing planning, each factory will update and transmit its encrypted data (e.g., material allocation information). The key generator is the only party that can decrypt the ciphertexts provided by the factories. Nonetheless, the supplier can solve the optimization problem without decrypting the data. Factory owners will then allocate materials to products accordingly. Overall, the entire system works in four phases.

Phase 1: Pricing

Fabrication and distribution pricing are decided when a supplier seeks to solve the optimization problem for all participating factories in the network. All participating factories will also evaluate all product backorder costs and communicate their total order quantities to the supplier.

Phase 2: Encryption

Each factory owner will encrypt their material allocation information before sending it to the supplier solving the optimization problem:

$$[\![x_j]\!] = E(x_j) \tag{28}$$

$$[\![z_k]\!] = E(z_k) \tag{29}$$

Phase 3: Secure manufacturing problem

The secure manufacturing problem is solved in a distributed manner over the ciphertexts according to Algorithm 1:

$$\min_{\{x_j\}_{j=1}^{J}} \sum_{k \in \mathcal{K}} C_k([\![z_k]\!]) + \alpha \sum_{j \in \mathcal{J}} U_j\left([\![x_j]\!]\right) \tag{30}$$

subject to

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} x_{j,k} \le \Omega \tag{31}$$

$$x_{j,k}^{\min} \le x_{j,k} \le x_{j,k}^{\max}, \ \ \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{32}$$

$$\sum\nolimits_{k \in \mathcal{K}} x_{j,k} = \Theta_j, \ \ \forall j \in \mathcal{J} \tag{33}$$

$$x_{j,k} = z_{j,k}, \ \ \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{34}$$

$$x_{j,k}, z_{j,k}, \ge 0, \ \ \forall \ j \in \mathcal{J}, k \in \mathcal{K} \tag{35}$$

The decision variables x and z are updated iteratively at the factory and supplier levels, respectively. Then, the dual variable μ is updated. The procedure continues until convergence to the optimal values.

However, ADMM minimization steps over ciphertexts have several complicating factors. Optimization is often carried out with the stochastic gradient descent algorithm. Paillier systems, in practice, require mantissa-based encoding to encrypt floating-point numbers. Mantissa-based encoding also provides significant gains in computational efficiency; without it, the Paillier cryptosystem is unusable on most machines at a reasonable level of security. Nonetheless, the usage of this type of mantissa-based encoding introduces the possibility of numeric overflow errors, which is a major pain point when attempting to perform gradient descent. Furthermore, the regularization term in the factory and supplier network update steps involves a squared Euclidean norm. The Paillier cryptosystem does not support the multiplication of two ciphertexts. Both issues can be mitigated through secure schemes involving the key generator. The key generator will fill the role of a trusted server, while the client role will be filled by either the factory or supplier network, as needed. The key generator is assumed to be trustworthy, but curious.

We propose Algorithm 2 to handle secure squaring of Paillier ciphertexts. In this context, the client desires to calculate a value $\llbracket x^2 \rrbracket$, but only has access to $\llbracket x \rrbracket$. Only the key generator possesses the private key which can decrypt $\llbracket x \rrbracket$. First, the client generates a random integer (r) and adds it to $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket = \llbracket x + r \rrbracket$. Doing this step blinds the value of x so that the key generator is not able to learn any detailed information about x. The ciphertext $\llbracket y \rrbracket$ is then sent to the key generator for decryption. The value of y^2 is calculated and then $\llbracket y^2 \rrbracket$ is sent back to the client. To obtain the desired $\llbracket x^2 \rrbracket$, the client calculates $\llbracket x^2 \rrbracket = \llbracket y^2 \rrbracket - 2r\llbracket x \rrbracket - r^2$.

Algorithm 2. Secure Squaring.

```
Input: $\llbracket x \rrbracket$
Output: $\llbracket x^2 \rrbracket$
with the client
    Generate random integer $r$
    Send $\llbracket y \rrbracket = \llbracket x + r \rrbracket$ to the key generator
end with
with the key generator
    Decrypt $\llbracket y \rrbracket$ to get $y$
    Compute $y^2$
    Send $\llbracket y^2 \rrbracket$ to the client
end with
Client calculates $\llbracket x^2 \rrbracket = \llbracket y^2 \rrbracket - 2r\llbracket x \rrbracket - r^2$
```

Performing minimization is a more involved process without gradient descent. The client seeks to determine the minimum of N encrypted values, $\min(\llbracket \delta_1 \rrbracket, \ldots, \llbracket \delta_N \rrbracket)$. In making this determination with the key generator, the client should not be able to derive any auxiliary information. For example, let the supplier network have ciphertext values $\llbracket 3 \rrbracket, \llbracket 17 \rrbracket$, and $\llbracket 5 \rrbracket$. The only information that should be knowable to the supplier network is that $\llbracket 3 \rrbracket$ is the lowest value among these three ciphertexts. Therefore, the ordering relationship among ciphertexts should be unknowable. Likewise, the key generator should not be able to derive any additional auxiliary information from the client when performing the decryption operation.

Thus, we propose Algorithm 3 to handle secure minimization of ciphertexts. First, the client generates a random permutation π of the encrypted values, giving us new ordering $\llbracket \delta_1' \rrbracket, \ldots, \llbracket \delta_N' \rrbracket$. The client will initialize itself by setting $\psi = \llbracket \delta_1' \rrbracket$ and $\ell^* = 1$. For iteration i, the client will then send $\llbracket \gamma \rrbracket = \psi - \llbracket \delta_i' \rrbracket$ to the key generator. Next, the client generates random integers r_0, r_1, which will be used to blind $\llbracket \delta_i' \rrbracket$ and ψ, respectively. Values $\llbracket \delta_i' + r_0 \rrbracket$ and $\llbracket \psi + r_1 \rrbracket$ are then sent to the key generator. The key generator will then decrypt $\llbracket \gamma \rrbracket$. Should $\gamma < 0$, then the key generator sets $\tau = 0$. Otherwise, it sets $\tau = 1$ and $\ell^\star = i$. The key generator will then compute three ciphertexts, whose purpose is to assist in the prevention of information leakage. These ciphertexts, $d_1 = \llbracket \min(\psi, \delta_i') + r_\tau \rrbracket$, $d_2 = \llbracket 1 - \tau \rrbracket$, and $d_3 = \llbracket \tau \rrbracket$ are then sent to the client. Using d_1, d_2, and d_3, the client will calculate $\psi = d_1 \cdot d_2^{-r_0} \cdot d_3^{-r_1}$. This algorithm continues until all N values are processed. The client will then obtain ℓ^* from the key generator, which will reveal the minimum value $\llbracket \delta^* \rrbracket = \pi^{-1}(\ell^*)$

Algorithm 3. Secure Minimization.

```
Input: $\llbracket \delta_\ell \rrbracket, \ell = 1, \ldots, N$
Output: $\llbracket \delta^* \rrbracket$
Client generates permuted ordering $\pi$, $\llbracket \delta_1' \rrbracket, \ldots, \llbracket \delta_N' \rrbracket$
Client sets $\psi = \llbracket \delta_1' \rrbracket$, $\ell^* = 1$
for $i = 1, \ldots, N$
    with the client
        Send $\llbracket \gamma \rrbracket = \psi - \llbracket \delta_i' \rrbracket$ to key generator
        Generate random integers $r_0, r_1$
        Send $\llbracket \delta_i' + r_0 \rrbracket, \llbracket \psi + r_1 \rrbracket$ to key generator
    end with
    with the key generator
        Decrypt $\llbracket \gamma \rrbracket$ to get $\gamma$
        if $\gamma < 0$ then
            $\tau = 0$
        else
            $\tau = 1$
            $\ell^* = i$
        end if
        $d_1 = \llbracket \min(\psi, \delta_i') + r_\tau \rrbracket$
        $d_2 = \llbracket 1 - \tau \rrbracket$
        $d_3 = \llbracket \tau \rrbracket$
        Send $d_1, d_2, d_3$ to client
    end with
    Client sets $\psi = d_1 \cdot d_2^{-r_0} \cdot d_3^{-r_1}$
end for
Key generator sends the client $\ell^*$
Client calculates $\llbracket \delta^* \rrbracket = \pi^{-1}(\ell^*)$
```

Theorem 2: *$d_1 \cdot d_2^{-r_0} \cdot d_3^{-r_1} = \llbracket min(\psi, \delta_i') \rrbracket$.*

Proof: First, suppose $\tau = 0$. We have, $\psi > \delta_i'$, and ciphertexts,

$$d_1 = \llbracket \delta + r_0 \rrbracket \; d_2 = \llbracket 1 \rrbracket, \text{ and } d_3 = \llbracket 0 \rrbracket$$

$$\begin{aligned} d_1 \cdot d_2^{-r_0} \cdot d_3^{-r_1} &= \llbracket \delta_i' + r_0 \rrbracket \cdot \llbracket 1 \rrbracket^{-r_0} \cdot \llbracket 0 \rrbracket^{-r_1} \\ &= \llbracket \delta_i' + r_0 \rrbracket \cdot \llbracket -r_0 * 1 \rrbracket \cdot \llbracket -r_1 * 0 \rrbracket \\ &= \llbracket \delta_i' + r_0 - r_0 - 0 \rrbracket \\ &= \llbracket \delta_i' \rrbracket, \text{ as desired.} \end{aligned}$$

Now, suppose $\tau = 1$. We have, $\psi < \delta_i'$, and ciphertexts,

$$d_1 = \llbracket \psi + r_1 \rrbracket \; d_2 = \llbracket 0 \rrbracket, \text{ and } d_3 = \llbracket 1 \rrbracket$$

$$\begin{aligned} d_1 \cdot d_2^{-r_0} \cdot d_3^{-r_1} &= \llbracket \psi + r_1 \rrbracket \cdot \llbracket 0 \rrbracket^{-r_0} \cdot \llbracket 1 \rrbracket^{-r_1} \\ &= \llbracket \psi + r_1 \rrbracket \cdot \llbracket -r_0 * 0 \rrbracket \cdot \llbracket -r_1 * 1 \rrbracket \\ &= \llbracket \psi + r_1 - 0 - r_1 \rrbracket \\ &= \llbracket \psi \rrbracket, \text{ as desired.} \end{aligned}$$

□

Phase 4: Decryption
Once the algorithm returns the optimal output, the result is decrypted. Products in each factory are then allocated materials accordingly.

4. Privacy analysis

The construction of distributed cryptosystem comes with inherent privacy properties in the manufacturing context. Furthermore, factory allocation information is protected against both suppliers in the network as well as the key generator, because the encryption method is homomorphic. Additionally, the system is resistant to attacks, due to the privacy-preserving Paillier indistinguishability property. These privacy features all coalesce to enable private, distributed optimization:

1. **Privacy protections from external threats:** Confidentiality and data privacy are achieved through the usage of Paillier encryption on the factory material allocation information. Thus, external eavesdroppers cannot learn anything by intercepting messages.
2. **Resistance to attacks:** The random value *r* which drives the Paillier encryption protocol provides ciphertext resilience against dictionary attacks. Multiple encryptions of a particular plaintext will yield ciphertexts that are statistically indistinguishable from each other.
3. **Privacy protection from supplier network and key generator:** The factory's private allocation information cannot be discerned by the supplier network nor the key generator. The supplier and key generator are assumed to be honest-but-curious, meaning that they follow all steps of their respective protocols correctly, but conduct data mining operations to learn information about other entities in the network. Factory allocations are kept private from the supplier network, because no intermediate information can be learned from the optimization protocol. Furthermore, although the key generator can decrypt information during the minimization and quantity squaring operations with the private key, the algorithms are designed so that the information is sufficiently obfuscated. Therefore, due to this obfuscation, the key generator cannot access any reliable information about the material allocations.
4. **Privacy-preserving ADMM:** The homomorphic nature of the Paillier cryptosystem enables computation over ciphertexts and drives privacy-preserving ADMM. The risk of information leakage is mitigated while solving the optimization problem.

5. Experimental design and materials

In this article, we evaluated and validated the performance of the distributed ADMM-cryptosystem framework against existing deterministic and stochastic methods to solve manufacturing planning problems. As shown in the experimental design in Figure 5, the number of factories and products are both varied from 5 to 30 (in increments of five) to accommodate different levels of supply chain complexity. Factories are assumed to keep inventory of both finished products and manufacturing materials.

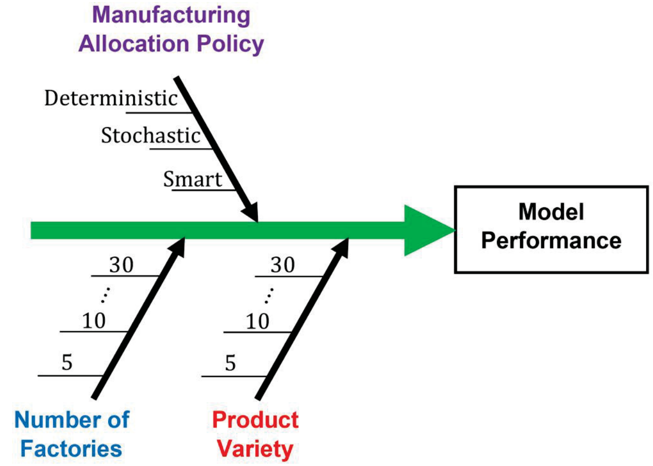

Figure 5. The cause-and-effect diagram for experimental evaluation of distributed manufacturing cryptosystem.

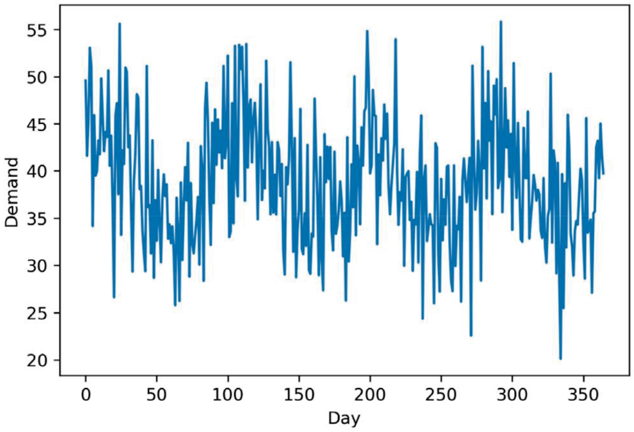

Figure 6. Simulated demand data with seasonality effects.

Material demand is assumed to follow a baseline profile that exhibits seasonality effects. As shown in Figure 6, the seasonality effect is generated from a sine wave with an amplitude of three and frequency 90. The remainder of the demand profile is generated from a Gaussian distribution with mean 30 and a standard deviation of three. Historical data are generated by perturbing this demand profile with Gaussian noise, utilizing a mean of zero and standard deviation of three. Altogether, 5 years of historical data are generated. A similar method is employed to generate test set data with the added effect of introducing a mean shift. One year of test set data is generated.

The material and production cost information are derived from real-world pricing data from Infineon Technology's stocks. The simulation framework tracks various additional costs to help discern model performance. Each time a factory places an order, a fixed cost of $10 is incurred. Additionally, the simulation considers material holding costs, which are set to be 10% of the material purchase cost over the course of a year. The supplier network distribution costs are set according to $\Lambda_j = 3 + (j-1),\ j = 1, ..., J$. Purchase cost is held to be a combination of distribution and production costs.

In addition to material costs, product backorder and product holding costs are traced. These costs follow $\beta_k = 10 \cdot k,\ k = 1, .., K$ and $h_k = 0.1 \cdot k,\ k = 1, ..., K$, respectively. Product inventory can be consumed to avoid product backorders.

Material order quantities are determined based on the manufacturing policy tested. The delivery of materials is assumed to have a static lead time of 3 days. The deterministic method utilizes an order schedule obtained from the Wager–Whitin algorithm. By contrast, the stochastic method utilizes an (s, S) order policy to decide both the reorder point and order quantity. A service level factor of two is employed when determining the reorder point.

Each factory must allocate materials to the fabrication of its products. Hence, product demand is modeled in the compositional fashion. Historical demand proportions are normally distributed within the K-dimensional simplex, centered on

$$\left[\frac{1}{\sum_{i=1}^{K} i}, ..., \frac{K}{\sum_{i=1}^{K} i}\right].$$

The deterministic and stochastic methods do not dynamically allocate materials to different products, but instead settle on the allocation scheme. The allocation is obtained from a weighted sum of the proportional demand lower bound (based on $x_{j,k}^{\min}$), upper bound (based on $x_{j,k}^{\max}$), and average. Deterministic and stochastic methods are used to benchmark the performance of the proposed distributed cryptosystem architecture in manufacturing planning and material allocations.

In addition to comparing price disparities, the convergence performance of the distributed cryptosystem will also be evaluated by assessing the objective function as the number of ADMM iterations increases using a relative error measure. Likewise, the computation efficiency of homomorphic encryption is shown against standard encryption. In the comparison experiments, we test a total of 10,000 scenarios where randomly generated numbers (between 0 and 500) undergo addition, subtraction, scalar multiplication, and scalar division, which encapsulates all of the possible operations supported under the Paillier cryptosystem.

6. Experimental results

This section presents the results of benchmark experiments to evaluate the performance of the proposed distributed cryptosystem for smart manufacturing. Figure 7 illustrates the computation times of mathematical operations (addition, subtraction, scalar multiplication, and scalar division) conducted under traditional and homomorphic encryption schemes. In each case, homomorphic encryption is significantly faster than standard encryption.

Figure 8(a) features the variation of ADMM convergence curves showing the relative error of the current allocation against the number of iterations performed. Different curves are shown for various factory user quantities $(J = 5, 10, ..., 30)$ while the number of products per user is kept static at $K = 5$. It is worth noting that the ADMM optimization program converges faster as the number of factory users in the system increases, albeit with diminishing returns. Figure 8(b) also shows the variations of ADMM convergence curves with respect to the number of products. Different curves are shown for various product quantities $(K = 5, 10, ..., 30)$ while the number of factory users in the system is kept static at $J = 5$. The ADMM optimization program converges at an increasingly slower rate as the number of products increases. Due to this negative impact on convergence performance, the number of ADMM iterations must be increased proportionally as K increases.

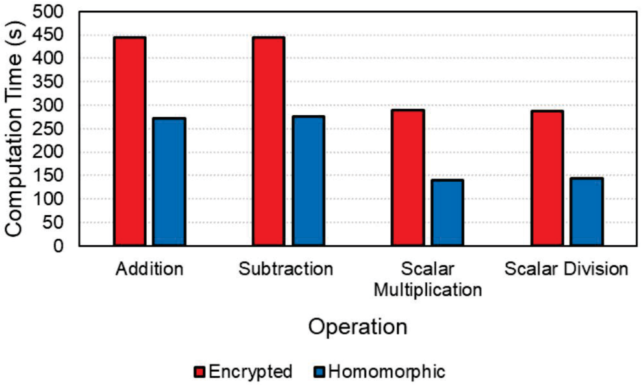

Figure 7. Comparison of computation times (in seconds) of mathematical operations conducted under traditional and homomorphic encryptions.

Furthermore, we have tracked and compared various cost metrics while assessing different policies for manufacturing planning. Figure 9(a) shows the breakdown of various cumulative costs incurred under a smart manufacturing policy with $J = 5$ factories, each with $K = 5$ products over a year. The material holding and order costs have minimal impact on the total cost of the policy and appear to overlap each other. For the majority of the year, the material purchase cost represents the largest cost, until the product holding cost overtakes at the end of the year. The product backordering costs lie under the product holding costs for the second half of the year. For the first half of the year, they are greater, albeit only by a small amount.

Figure 9(b) shows a comparison of cumulative costs of the deterministic, stochastic manufacturing policies against the proposed distributed cryptosystem over a year. The deterministic method incurs the greatest cost for the entire year. The smart policy of the proposed distributed cryptosystem incurs the smallest cost over the year. The stochastic method incurs costs that lie in-between the deterministic policy and the proposed distributed cryptosystem. However, the stochastic (s, S) policy costs start catching up to the deterministic policy costs towards the year's end.

Figure 10(a) shows a comparison of the total yearly costs between deterministic, stochastic policies, and the proposed distributed cryptosystem as the number of factory users increases in the network. The number of products per user is kept static at $K = 5$. It should be noted that the deterministic method always operates at the highest cost while the

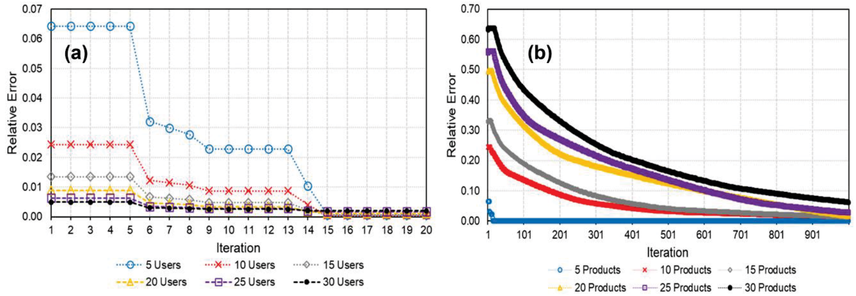

Figure 8. (a) ADMM convergence curves, varying the number of users ($K = 5$) and (b) ADMM convergence curves, varying the number of products ($J = 5$).

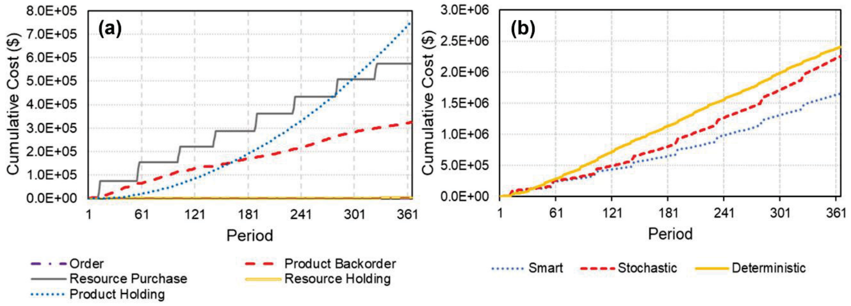

Figure 9. (a) Cost ($) breakdown under smart manufacturing policy over a year ($J = 5, K = 5$) and (b) cost ($) comparison between manufacturing policies over a year ($J = 5, K = 5$).

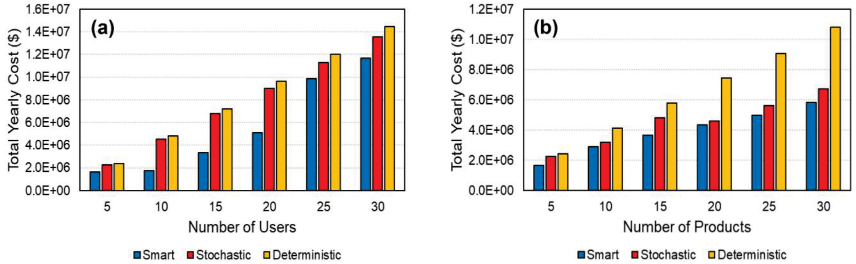

Figure 10. (a) Total yearly cost ($) of manufacturing policies as the number of users is varied, but the number of products is fixed ($K = 5$) and (b) total yearly cost ($) of manufacturing policies as the number of products is varied, but the number of factories is fixed ($J = 5$).

proposed distributed cryptosystem always operates at the lowest cost. The stochastic method lies between the two, though tends to closely trail the incurred costs of the deterministic method. Nonetheless, there is a sudden jump in total yearly cost between the "20-user" and "25-user" configurations for the smart method.

By contrast, Figure 10(b) shows a comparison of the total yearly costs between deterministic, stochastic policies, and the proposed distributed cryptosystem as the number of products per factory increases. Note that the deterministic method is the most expensive in all cases. However, it greatly outpaces the costs incurred by both the stochastic policy and the proposed distributed cryptosystem as the number of products increases. The proposed distributed cryptosystem yields the least cost in all cases, but closely trails the costs of the stochastic (s, S) policy.

7. Conclusions

Smart manufacturing is increasingly susceptible to the ever-looming threat and impact of cyber-attacks. The new reality of large-scale data proliferation thus solidifies reliance on distributed storage and computation, leaving service-oriented systems vulnerable to a myriad of privacy and security risks. As a result, there is an urgent need to diminish the risk of a data breach. In this investigation, we propose a new distributed cryptosystem for smart manufacturing. The smart manufacturing optimization program allows for materials to be allocated to products in a cost-effective manner. This cost effectiveness is achieved through coordination between factories and a supplier network. On the whole, the smart allocation methodology achieves a better performance than benchmark deterministic and stochastic policies under different experimental configurations. Convergence efficiency of the proposed method is largely dependent on the number of factory users and number of products. Convergence is generally better as the number of users increases and is generally degrading as the number of products increases.

Future work will entail the exploration of alternative cryptosystems for solving the smart manufacturing problem. The Paillier cryptosystem does not support multiplication of ciphertexts nor is it quantum resistant. Fully homomorphic cryptosystems may help further reduce computational overhead. Likewise, quantum computers are advancing at an accelerated pace and may leave existing cryptosystems vulnerable to Shor's algorithm-based attacks. Furthermore, future work will tackle expanding the smart manufacturing problem to handle multiple material types and tiered supply chain systems.

The distributed cryptosystem framework that empowers the decentralized coordination for manufacturing planning comes with a myriad of security features. These features include adversarial privacy protections, resistance to attacks, privacy protection from honest-but-curious third parties, and privacy-preserving ADMM. These features stem from the homomorphic nature of the Paillier cryptosystem. Overall, distributed cryptosystem enables computation on encrypted data and drives the generation of secure and robust analytical insights in the smart manufacturing domain.

Funding

The authors of this work would like to acknowledge the NSF grants IIS-2302834 and MCB-1856132 for funding this research. Any opinions, findings, or conclusions found in this paper are those of the authors and do not necessarily reflect the views of the sponsors. National Science Foundation

Notes on contributors

Dr. Alexander Krall is currently an assistant research professor in the Materials and Manufacturing Office at the Applied Research Laboratory, The Pennsylvania State University. He was a PhD student in the Complex System Monitoring, Modeling and Control laboratory at the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University. He received both his Bachelor of Science (2016) and Master of Science (2018) degrees in industrial & systems engineering at the Rochester Institute of Technology. Alexander's primary research areas are distributed security, differential privacy, and quality-driven data analytics pertinent to complex manufacturing and healthcare systems.

Dr. Daniel Finke is an associate research professor in the Materials and Manufacturing Office at the Applied Research Laboratory, The Pennsylvania State University and the director of the Center for e-Design. Much of Dr. Finke's experience in applied research and development is within the US Navy shipbuilding domain collaborating on projects in Advanced Manufacturing Enterprise with a focus on production and capacity planning, Industrial Internet of Things (IIoT), and manufacturing system modeling and analysis. Dr. Finke received his PhD in industrial engineering (2010) and MS in industrial engineering and operations research (2002) from the Pennsylvania State University and a BS in industrial engineering from New Mexico State University (2000). His current research interests include simulation-based decision support, planning and scheduling, heuristic algorithm development and implementation, agent-based simulation and modeling, and process improvement.

Dr. Hui Yang is a professor of industrial and manufacturing engineering at The Pennsylvania State University, University Park, PA. He is the director of Complex System Monitoring, Modeling and Control laboratory. His research interests are sensor-based modeling and analysis of complex systems for process monitoring, process control, system diagnostics, condition prognostics, quality improvement, and performance optimization. He received the NSF CAREER award in 2015, and multiple best paper awards from the international IEEE, IISE and INFORMS conferences. Dr. Yang is the president (2017–2018) of IISE Data Analytics and Information Systems Society, the president (2015–2016) of INFORMS Quality, Statistics and Reliability (QSR) society, and the program chair of 2016 Industrial and Systems Engineering Research Conference (ISERC). He is also the Editor-in-Chief for *IISE Transactions Healthcare Systems Engineering*, as well as associate editors for *IISE Transactions, IEEE Journal of Biomedical and Health Informatics (JBHI), ASME Journal of Computing and Information Science in Engineering (JCISE).*

ORCID

Alexander Krall http://orcid.org/0000-0002-9753-1523
Daniel Finke http://orcid.org/0000-0001-5370-0412
Hui Yang http://orcid.org/0000-0001-5997-6823

References

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C.A. and Strand, M. (2015) A guide to fully homomorphic encryption. *IACR Cryptol. EPrint Arch.*, *2015*, 1192.

Balta, E.C., Lin, Y., Barton, K., Tilbury, D.M. and Mao, Z.M. (2018) Production as a service: A digital manufacturing framework for optimizing utilization. *IEEE Transactions on Automation Science and Engineering*, **15**(4), 1483–1493.

Biehler, M., Zhong, Z. and Shi, J. (2024) SAGE: Stealthy attack generation in cyber-physical systems. *IISE Transactions*, **56**, 54–68.

Bourne, V. (2022) The state of industrial security in 2022, Market Report, Barracuda Networks, Campbell, California.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, **3**(1), 1–122.

Errapotu, S. M., Wang, J., Gong, Y., Cho, J.-H., Pan, M. and Han, Z. (2018) SAFE: Secure appliance scheduling for flexible and efficient energy consumption for smart home IoT. *IEEE Internet of Things Journal*, **5**(6), 4380–4391.

Hopp, W.J. and Spearman, M.L. (2008) *Factory Physics* (3rd ed.), Waveland Press, Inc, Long Grove, IL.

Huo, X. and Liu, M. (2022) Privacy-preserving distributed multi-agent cooperative optimization—paradigm design and privacy analysis. *IEEE Control Systems Letters*, **6**, 824–829.

Kan, C., Yang, H. and Kumara, S. (2018) Parallel computing and network analytics for fast Industrial Internet-of-Things (IIoT) machine information processing and condition monitoring. *Journal of Manufacturing Systems*, **46**, 282–293.

Khojasteh, Y. (2016) Production systems, in *Production Control Systems: A Guide to Enhance Performance of Pull Systems* Springer Japan, pp. 7–24.

Krall, A., Finke, D. and Yang, H. (2021) Mosaic privacy-preserving mechanisms for healthcare analytics. *IEEE Journal of Biomedical and Health Informatics*, **25**(6), 2184–2192.

Kraning, M., Chu, E., Lavaei, J. and Boyd, S. (2014) Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, **1**(2), 73–126.

Lee, H., Finke, D. and Yang, H. (2023) Privacy-preserving neural networks for smart manufacturing. *Journal of Computing and Information Science in Engineering*, 1–20. doi: 10.1115/1.4063728

Lee, H. and Yang, H. (2023) Digital twinning and optimization of manufacturing process flows. *Journal of Manufacturing Science and Engineering*, **145**(11), 111008-1–111008-13.

Li, D., Paynabar, K. and Gebraeel, N. (2021). A degradation-based detection framework against covert cyberattacks on SCADA systems. *IISE Transactions*, **53**(7), 812–829.

Li, N., Zhou, T., Yang, X., Han, Y. and Sun, Y. (2018) Efficient fully homomorphic encryption with large plaintext space. *IETE Technical Review*, **35**(sup1), 85–96.

Liu, Y., Wang, L., Wang, X.V., Xu, X. and Jiang, P. (2019) Cloud manufacturing: Key issues and future perspectives. *International Journal of Computer Integrated Manufacturing*, **32**(9), 858–874.

Miller, A., Hartman, N., Hedberg, T., Barnard Feeney, A. and Zahner, J. (2017) Towards Identifying the Elements of a Minimum Information Model for Use in a Model-Based Definition, in *Proceedings of the ASME 2017 12th International Manufacturing Science and Engineering Conference MSEC2017*, June 4–8, 2017, Los Angeles, CA, USA, pp. 1–13.

Paillier, P. (1999) Public-key cryptosystems based on composite degree residuosity classes, in *Advances in Cryptology—EUROCRYPT '99*, Springer, Berlin, Heidelberg, pp. 223–238.

Si, F., Zhang, N., Wang, Y., Kong, P.-Y. and Qiao, W. (2023) Distributed optimization for integrated energy systems with secure multiparty computation. *IEEE Internet of Things Journal*, **10**(9), 7655–7666.

Stouffer, K., Zimmerman, T., Tang, C.Y., Pease, M., Cichonski, J. and McCarthy, J. (2020) *Cybersecurity Framework Version 1.1 Manufacturing Profile*, National Institute of Standards and Technology, Gaithersburg, Maryland.

Vedeshin, A., Dogru, J.M.U., Liiv, I., Ben Yahia, S. and Draheim, D. (2020) A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. *IEEE Access*, **8**, 40039–40056.

Wu, T., Zhao, C. and Zhang, Y.-J.A. (2021) Privacy-preserving distributed optimal power flow with partially homomorphic encryption. *IEEE Transactions on Smart Grid*, **12**(5), 4506–4521.

Yang, H., Bukkapatnam, S.T. and Barajas, L.G. (2013) Continuous flow modelling of multistage assembly line system dynamics. *International Journal of Computer Integrated Manufacturing*, **26**(5), 401–411.

Yang, H., Chen, R. and Kumara, S. (2021) Stable matching of customers and manufacturers for sharing economy of additive manufacturing. *Journal of Manufacturing Systems*, **61**, 288–299.

Yang, H., Kumara, S., Bukkapatnam, S.T.S. and Tsung, F. (2019) The internet of things for smart manufacturing: A review. *IISE Transactions*, **51**(11), 1190–1216.

Ye, Z., Cai, Z., Yang, H., Si, S. and Zhou, F. (2023) Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning. *Reliability Engineering & System Safety*, **236**, 109290.

Ye, Z., Yang, H., Cai, Z., Si, S. and Zhou, F. (2021) Performance evaluation of serial-parallel manufacturing systems based on the impact of heterogeneous feedstocks on machine degradation. *Reliability Engineering & System Safety*, **207**, 107319.

Zhang, C., Ahmad, M. and Wang, Y. (2018) ADMM based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, **14**(3), 565–580.