Secure Blockchain-Based Supply Chain Management with Verifiable Digital Twins

Vincenzo Botta*  
v.botta@uw.edu.pl  
University of Warsaw  
Warsaw, Poland

Attilio Mondelli  
Farzati Tech.  
Casal Velino, Italy  
attilio.mondelli@farzatitech.it

Laura Fusco†  
laurafusco1995@gmail.com  
University of Salerno  
Fisciano, Italy

Ivan Visconti  
visconti@unisa.it  
University of Salerno  
Fisciano, Italy

ABSTRACT
A major problem in blockchain-based supply chain management is the potential unreliability of digital twins when considering digital representations of physical goods. Indeed, the use of blockchain technology to trace goods is obviously ineffective if there is no strong correspondence between what is physically exchanged and the digital information that appears in blockchain transactions.

In this work, we propose a model for strengthening the supply chain management of physical goods by leveraging blockchain technology along with a digital-twin verification feature. Our model can be instantiated in various scenarios and we have in particular considered the popular case of food traceability. In contrast to other models known in the literature that propose their own ad-hoc properties to assess the robustness of their supply chain management systems, in this work, we use the formalism of secure computation, where processes are described through generic and natural ideal functionalities.

CCS CONCEPTS
- Security and privacy → Distributed systems security.

KEYWORDS
Blockchain, Supply Chain, Digital Twins, Secure Computation

ACM Reference Format:
Vincenzo Botta, Laura Fusco, Attilio Mondelli, and Ivan Visconti. 2023. Secure Blockchain-Based Supply Chain Management with Verifiable Digital Twins. In ACM International Conference on Information Technology for Social Good (GoodIT ’23), September 06–08, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3582515.3609547

INTRODUCTION
The traceability of goods is extremely important to optimize their management and to reduce risks of frauds and their consequences (e.g., in foodstuffs frauds damage the economy and public health). Mutually distrustful supply chain (SC) members are interested in using tracing systems since traceability provides transparency and helps to fight counterfeiting with positive effects on the reputation and business of the involved companies [23].

Traditional supply chain management systems are centralized and their credibility relies on the trust towards central administrators [3, 22]. This single point of failure due to centralized architectures is a classical issue that motivates the use of blockchain technology. Indeed, blockchains can help to provide an immutable, consistent, transparent, decentralized and highly available ledger of supply chain data. Blockchain technology is nowadays mature enough to be used in scenarios decentralizing classic centralized solutions at low cost and high transaction rate when using their governance is restricted (i.e., when using a permissioned blockchain). By leveraging blockchain technology, all events in a supply chain can be digitized and stored in a transparent and immutable manner with robust data availability. Blockchain-based solutions provide the traceability of the supply chain by recording the status of a good at each stage of the supply chain.

Achilles’ heel: the validity of a digital twin. A supply chain management system provides ways to connect a virtual digital good to a concrete physical good [17]. Unfortunately, there is a major issue to consider: the problematic connection between data stored in a blockchain and the actual goods in the physical world [18]. Consider for instance a farm producing mozzarella with milk obtained from Italian Mediterranean buffalo. A blockchain is useful to transparently give information on the location and size of the farm, the number of buffaloes and the amounts of buffalo mozzarella that are daily produced and reach the stores and the final consumers. Notice that a producer may be interested in counterfeiting buffalo mozzarella, which is a protected food, by using cow’s milk to increase profits. The use of a blockchain allows to publicly verify that the farm does not claim unreasonably high amounts of produced buffalo mozzarella. While this tracing is beneficial for the entire system, it still remains possible to dishonestly mix buffalo milk and cow milk whenever there is a shortage of buffalo milk, due for instance to a virus affecting...
buffaloes. This is a typical case where while data appearing in the blockchain are consistent with expectations, they can be completely uncorrelated to the physical goods to which they refer to.

1.1 Our Contribution

In this paper, we present a general model for blockchain-based supply chains focusing on the traceability and the verification of goods. Our goal is to enable every party in the system to retrieve full information about the life-cycle of goods in the supply chain. Each physical good has a digital twin identified by a unique code that appears on a blockchain along with all the relevant events describing the life of the good. A reliable auditing process is required in order to strengthen consistency between the information stored on-chain and the physical goods, therefore reducing errors and frauds. Every time an inspection is carried out by a supply chain entity some relevant information (e.g., date, time, place, operator, verified goods, the outcome of the inspection) about the inspected good is recorded on the blockchain.

We formalize the concept of reliable audit through a specific ideal functionality that can be instantiated in several ways. In particular, our model focuses on establishing how to declare a category of goods that will be involved in the supply chain and then how to make sure that goods that are supposed to belong to a certain category can be tested in case of inspection. Notice that defining this process is non-trivial given the evolving nature of goods during their life-cycles (e.g., the color, hardness and flavor of banana are expected to change day by day depending also on how they are stored). When presenting our model we will be generic about the goods, but for concreteness, we will sometimes refer in particular to food traceability. Indeed, this use case represents a popular example where the use of blockchains has been widely considered and bogus digital twins can severely affect desired advantages of publicly verifiable tracing. The formalism that we use to guarantee the robustness of the traceability provided by our model is the gold standard in the design of secure systems: secure multi-party computation (SMPC) [8]. It consists of defining a natural ideal world representing how traceability can be reliably performed in the presence of trusted parties. Then, the reliability of a concrete real-world system can be measured by showing that any attack on the concrete system can be (through a simulation) carried also in the above ideal world. Since by inspection one can easily check the reliability of processes run in the ideal world, and since the real-world system is proven as secure as executions in the ideal world, one can conclude that the proposed real-world system is reliable too.

For lack of space in this version we omit some details (e.g., formal descriptions of functionalities, security proofs, formal definitions) that are available in the full version [4].

2 RELATED WORK

Our work mainly focuses on designing a framework that combines a permissioned blockchain with a reliable auditing process that together make possible the traceability and the verification of goods in a supply chain.

The concept of a digital twin was discussed by Grieves [10]. Later on, many works have summarized and reviewed digital twin technologies and their applications. Jones et al. [13] and Marmolejo-Suácedo et al. [20] presented literature reviews reporting various limitations in the state of the art, and remaking that the connection between physical and virtual goods is still a main challenge. Similarly, Melesse et al. [17] studied the inaccuracy of the digital representation of a good or asset with respect to its physical twin. Deng et al. [7] reviewed works on digital twin applications in urban governance and identified blockchains as a tool to guarantee the security and ownership of goods. Tebaldi et al. [21] presented preliminary results on digital twins in the context of the food supply chain to assess the state of the art of this technology. Jabbar et al. [12] discussed many aspects of supply chain management with blockchains. Defraeye et al. [6] identified key advantages of digital twins in the context of fresh horticultural products. Rivera et al. [19] proposed a model to specify structural and procedural aspects of digital twins. Yaqoob et al. [24] and Liu et al. [16] analyzed the benefits of employing blockchains in platforms managing digital twins. However, their work did not discuss any procedure to check the consistency of the information stored in the chain. Guo et al. [11] investigated the challenge of managing personalized services for products with a limited life-cycle. They focused on designing a system in which customers can participate and are involved in improving production. However, in their work, the auditing process is not formalized. Lee et al. [15] focused on accountable information sharing among participants in the digital twin system. Moreover, Lee et al. have indicated the use of permissioned blockchains like Hyperledger Fabric [1] as a future research possibility focusing also on the need of combining on-chain and off-chain data.

3 SYSTEM OVERVIEW

Our blockchain-based tracing system allows supply chain participants to record information on goods at each stage of the supply chain. Each physical good has a unique digital twin that is identified by an asset on the blockchain. The asset is associated with a current owner, a list of events and some auxiliary information. The owner of an asset identifies the party that is supposed to own the physical good. The events represent the life-cycle of the physical good and the auxiliary information gives additional information on the asset (e.g., location, type of good).

In our system, access-control policies are defined to prevent non-legitimate participants, including counterfeiters, from jeopardizing the ownership of assets. Moreover, access-control policies guarantee that only legitimate manufacturers can claim the initial ownership of new assets that are introduced in the system, and that are supposed to correspond to physical goods. The creation of an asset is possible only if the party invoking this functionality has an legitimate role (e.g., it is enrolled in the system as a farmer).

Our system allows a party to verify the authenticity of received physical goods by comparing them with their digital twins. This verification process is abstracted by assuming the existence of a device called fingerprint scanner (FP scanner), which can identify whether a given object belongs to a particular category after
being trained\(^1\). The capability to validate a digital twin affects the enrollment of categories of goods in the system.

**System set-up and assets.** At first, a genesis block is created to fix some system parameters, access-control policies and system smart contracts. After the system initialization, supply chain participants (e.g., producers, distributors, certifiers) engage in an enrolling protocol to get authorizations for their subsequent interactions. From now on, parties can record information in the system by submitting transactions.

In our system, we have five different types of assets: (1) **element**, a good, that can be raw, or the result of a transformation; (2) **category**, a category to which goods can belong; (3) **batch**, a batch of goods created aggregating other goods; (4) **production area**, a description of the location in which goods are produced; (5) **device**, a scanner associated with a party that is used to analyze the fingerprint of the products, that we call FP scanner.

**Entities.** There are seven types of participants in our system.

- **Registration authorities.** They are privileged parties that initialize and govern the system by registering participants with long-term credentials. The access control policy (i.e., who is allowed to add a specific transaction) and the logic of smart contracts are decided by these organizations.
- **FP scanner issuers.** They have the exclusive authority to release new FP scanners to ensure a desired level of service quality. Parties must request an FP scanner from an issuer, and registration authorities establish criteria for adding or removing issuers. The list of authorized FP scanner issuers must be publicly available.
- **Producers.** They are authorized to introduce new items consisting of raw materials and record production areas. They also link physical goods to digital twins on the blockchain to track their supply chain journey. Producers can create categories, define properties of items and update information.
- **Manufacturers.** They are authorized users in charge of producing goods for sale from raw materials. They can introduce new items obtained either from raw materials or from the transformation of existing goods. They can create new categories and benchmarks and register the production area used to transform their goods. Additionally, manufacturers have access to all operations allowed to supply chain members.
- **Certifiers.** They are entities responsible for certifying the authenticity of the physical good with respect to the information encoded on-chain. An audit recorded by a certifier is different from an audit recorded by other parties. Certification bodies are entities authorized by accreditation bodies to conduct specific types of certification audits and establish certifications for qualified companies. The certifiers verify if the goods, the systems, the personnel, and the production processes of companies satisfy the certification requirements.

- **Consumers.** They can access (read-only) the full history of items that are managed by the supply chain. They can also be involved in providing feedback on purchased items as active participants in the supply chain.
- **Supply chain members.** They are the remaining participants of the supply chain, (e.g., wholesaler, retailer, distributor, repackager). They are grouped into a single category since they can perform the same operations. These entities own items and give/receive items or aggregate/disaggregate them in larger/smaller packages and/or update data corresponding to items.

**Trust assumptions.** For the sake of simplifying the description of our model, we will consider a single trusted registration authority. The registration authority Reg is trusted to assign correct credentials to all parties in the system and one unique identity to each participant. Moreover, Reg is trusted to verify the correctness of the attributes of a participant before enrolling the user in the system and is also trusted in verifying the veracity of the role declared by parties. Reg can be obviously decentralized, as described in Section 6.3. Moreover, we will consider a single FP scanner issuer \( D \), (i.e., the producer of the devices). We assume that \( D \) issues only devices that are well-formed and thus implement the corresponding functionality correctly. This trust assumption can be relaxed, as described in Section 6.3. Producers and Manufacturers are trusted to introduce new categories and/or update existing categories since such procedure are performed under trusted supervision. Moreover, we assume that an adversary cannot compromise data stored in the blockchain, and cannot prevent transactions from being added to the ledger.

## 4 SECURITY MODEL

We follow the simulation paradigm considering ideal and real worlds using the notion of universally composable (UC) secure multi-party computation [5]. We consider hybrid models in which protocols access ideal functionalities to perform the computation. Informally, a protocol \( \pi \) is executed in the \( G \)-hybrid model if \( \pi \) invokes the ideal functionality \( G \) as a subroutine. Let \( F \) be a functionality, and consider \( n \) players \( P_1, \ldots, P_n \) executing a protocol \( \pi \) that implements \( F \). \( \pi \) is secure if, in an execution in the real world, the adversary cannot cause more harm than an adversary in the ideal world in which a trusted party computes the function corresponding to the ideal functionality \( F \) on behalf of the players. We require that this guarantee holds even if \( \pi \) accesses a functionality \( G \) as a subroutine.

In the real world, the protocol \( \pi \) is run in the presence of an adversary \( A \) and an environment \( Z \). \( Z \) chooses the inputs for each honest player \( P_i \), and gives in input to \( A \) an auxiliary input \( z \) and inputs for corrupted parties. We only consider static corruption. Honest players \( P_i \) run \( \pi \) behaving as prescribed in the protocol, while corrupted parties behave arbitrarily driven by \( A \). At the end of the execution, \( Z \) receives the output of honest and corrupted parties and outputs a bit.

In the ideal world, there exists a trusted party that executes the ideal functionality \( F \) on behalf of the dummy players. \( Z \) chooses the inputs for each honest player and sends to an adversary \( S \), called the simulator, inputs of corrupted parties together with

---

\(^1\)An example of such a device for food traceability is a biological fingerprint scanner that can distinguish between different types of mozzarella.
an auxiliary input $z$. Honest parties send their inputs to the trusted party, while the corrupted parties send to the trusted party data computed as specified by $S$. The trusted party executes the functionality and produces outputs that are given to parties. Finally, $S$ computes an arbitrary function of inputs received initially by $Z$ and outputs received by the trusted party and gives the output of the computation to $Z$ that returns a bit.

UC-Secure MPC. Let $\pi$ be an $n$-party protocol that implements a functionality $F$. Informally, $\pi$ UC-securely realizes $F$ in the $G$-hybrid model in the presence of malicious adversaries if for every PPT adversary $A$ there exists a PPT simulator $S$ such that for every non-uniform PPT environment $Z$ the executions in real and the ideal worlds are computationally indistinguishable.

5 THE IDEAL FUNCTIONALITIES

In this section, we define the ideal functionality realized by our system for product tracing $F_{SC}$. $F_{FP}$ models the FP scanner that parties can use to test an item. For lack of space, we will instead omit the formal descriptions of the following more standard functionalities: $F_{L}$ledger that emulates a transaction ledger that parties can use to submit transactions or to read the current state of an item; $F_{Sig}$ that models a secure digital signature scheme allowing each party to sign digital messages and verify digital signatures; $F_{SMR}$ that models a secure channel required to privately and securely exchange data between parties.

5.1 Supply-Chain Functionality $F_{SC}$

Since in the real world each supply-chain member, according to the assigned role, can perform different operations, $F_{SC}$ requires that producers $F$, manufacturers $M$, certifiers $C$, and other SC-members $O$ are enrolled in the system. be registered.

A party $P_i$ that would like to characterize a category of products $p$ will send to $F_{SC}$ a value $\text{catFP}$ representing the fingerprint of $p$. $\text{catFP}$ can be computed for example using the $F_{FP}$ functionality presented below. A party $P_i$ that wants to send the outcome of an audit of a product $p$ to $F_{SC}$, will send a value $\text{auditData}$ to $F_{SC}$: representing data of the audit. This value can be computed leveraging $F_{FP}$.

Each product has a state whose possible values are: (i) intact: meaning that the asset is ready for use; (ii) packaged: meaning that the asset has been packaged and is not ready to be used individually. A disaggregation event must occur before becoming available again (batch operations have a collective impact on the entire batch, therefore, it is essential to perform disaggregation before conducting any operation on an item within the batch); (iii) trans: indicates that a handover has started between two parties; (iv) destroyed: indicates that the asset is no longer available for future processing.

For simplicity we implicitly assume that any information sent to $F_{SC}$ is verified through the following checks that can be implemented by inspecting data stored in $F_{SC}$: (i) the stored product complies with rules, regulations, and standards; (ii) producers and manufacturers own the production area declared; (iii) the fingerprint $\text{catFP}$ produced during a training activity on a product can be updated only by the party that created it; (iv) identifiers of categories, production areas, and products are unique; (v) a producer or manufacturer creating a new product assigned to a production are must own the production area and all needed items at creation time; (vi) parties perform operations only over items on which they have the rights to operate; (vii) aggregation and disaggregation operations can be performed only on owned products; (viii) the items to be packaged in an aggregation operation must be intact; (ix) the disaggregation operation can be performed only on an intact batch and the products in the batch must be packaged; (x) the update event of an item in the supply chain is only executed if the requesting party is the current owner of the product and only for specific fields of information; (xi) the party that performs a handover is the owner of the good to transfer and the good is intact; (xii) the destination party for a handover is authorized to receive the asset; (xiii) the goods that parties transfer in a handover are recorded goods.

$F_{SC}$ uses an algorithm $\text{Transform}$ that informally transforms a product or a set of products into a new product. This algorithm is used by manufacturers to transform existing assets into new assets. $\text{Transform}$ modifies the state of the existing assets from intact to destroyed.

We call $\text{catFP}$ the category associated with a fingerprint that uniquely identifies a type of asset. Data stored by $F_{SC}$ are associated with a timestamp $tEvent$. $F_{SC}$ models the sequence of events that occurs along a supply chain. Each event corresponds to an operation that a party can perform. The results of the operations are stored in an initially empty list $L$.

A particular case of an update is deleting a product instance from the supply chain. A deletion event is essentially the inverse of a creation event: once deleted, an asset is invalidated and can no longer be used. It can occur when products are sold to the end-users or there is a product recall or loss due to contamination or accident. In case of a product recall, competent authorities will pick the product up from the market.

$F_{SC}$ Ideal Functionality

$F_{SC}$ stores an initially empty list $L$. Parties $P_1, \ldots, P_n$ and adversary $A$ that interact with $F_{SC}$ may be of type $F, M, C, O$ (i.e. producers, manufacturers, certifiers, and others).

- Registration of a party. Whenever a party writes $(\text{register}, P_i)$, if $P_i$ is unregistered, then $F_{SC}$ marks $P_i$ as registered and outputs $(\text{registered}, P_i)$ to $P_i$ and $A$. Else, $F_{SC}$ sends $\bot$ to $P_i$.
- Registration of a production area. To record the place of production of a new product a party $P_i \in \{F, M\}$ writes $(\text{produce}, P_i, \text{areaId}, \text{categId})$. $F_{SC}$ stores $(P_i, \text{areaId}, \text{categId}, \text{producer}, tEvent, \text{intact})$ in $L$. $F_{SC}$ sends $(\text{produced}, P_i, \text{areaId})$ to $P_i$ and $A$.
- Creation. To create a new product a party $P_i \in \{F, M\}$ writes $(\text{create}, P_i, \text{itemId}, \text{areaId}, \text{categId})$. $F_{SC}$ stores $(P_i, \text{itemId}, \text{areaId}, \text{creation}, tEvent, \text{intact})$ in $L$. $F_{SC}$ sends $(\text{created}, P_i, \text{itemId}, \text{areaId})$ to $P_i$ and $A$.
- Transformation. To create a new product instance through the combination of single or multiple products a party $P_i \in M$ writes $(\text{transform}, \text{itemId}, P_i, \text{items}, \text{categId})$, where $\text{items} = \{\text{itemId}_0, \ldots, \text{itemId}_n\}$. $F_{SC}$ checks that the state of each item is intact. $F_{SC}$ runs $\text{Transform}(\text{items})$, storing in $L$ a new tuple for each itemId $\in \text{items}$ where the state passes from intact to destroyed. Finally, $F_{SC}$ stores $(P_i, \text{itemId}, \text{items}, \text{itemId}_0, \text{categId}, \text{transformation}, tEvent, \text{intact})$ in $L$. $F_{SC}$ sends $(\text{transformed}, P_i, \text{itemId}, \text{categId}, \text{items})$ to $P_i$ and $A$.
- Training. To perform a training activity a party $P_i \in \{F, M\}$ writes $(\text{train}, P_i, \text{categId}, \text{catFP})$. $F_{SC}$ stores
Each new scanner has associated a tuple \((P_i, deviceId)\) (i.e., the scanner with identifier deviceId belongs to \(P_i\)). The device can be transferred from one party to another. In this case, we assume that once the handover happens, the owner of the device is updated. We assume that every time that operations of \(FPF\) are called, the caller is the owner of the used device and in the case of a handover the device is not yet engaged in another handover procedure.

The \(FPF\) functionality includes procedures \(Training\), \(Verify\), and \(Sample\). \(Training\) generates a specific fingerprint, \(catFP\), that should represent the biochemical and molecular mapping of a trained category. It can be a time-consuming procedure that could make the trained products unusable. \(Verify\) is used to determine if a physical product belongs to a category. Compared to \(Training\), \(Verify\) is expected to have a shorter execution time and is less prone to make unusable the tested products. \(Sample\) is an efficient procedure that samples a univocal value. We require that the initial training and next updates can be performed only by the party that creates the category \(catFP\).

We make the following trust assumptions on the scanner: (i) the parties can verify the authenticity of the hardware (e.g., the integrity of the hardware)\(^2\); (ii) the FP scanner can be uniquely identified; (iii) the output of an FP scanner is associated with its identifier. Moreover, we assume that an adversary has access only to FP scanners possessed by corrupted parties.

### 5.2 AssetFingerPrint Scanner Functionality \(FPF\)

\(FPF\) handles the following operations.

- **Initialize a device.**
- **Perform the training operations to allow \(FPF\) to recognize a specific category of products.**
- **Verify that a given item belongs to the claimed category.**
- **Transfer a device from party \(P_i\) to party \(P_j\).**
- **List all FP scanners issued to parties.**

Each new scanner has associated a tuple \((P_i, deviceId)\) (i.e., the scanner with identifier deviceId belongs to \(P_i\)). The device can be transferred from one party to another. In this case, we assume that once the handover happens, the owner of the device is updated. We assume that every time that operations of \(FPF\) are called, the caller is the owner of the used device and in the case of a handover the device is not yet engaged in another handover procedure.

### 6 THE REAL-WORLD: OUR SYSTEM \(\Pi_{SC}\)

Our supply chain protocol \(\Pi_{SC}\) realizes the functionality \(\Pi_{SC}\) and operates with functionalities \(\Pi_{ledger}\), \(\Pi_{MT}\), \(\Pi_{Sig}\) and \(\Pi_{FP}\). \(\Pi_{ledger}\) models a transaction ledger and is parameterized by an algorithm to submit transactions and an algorithm to read the state of the blockchain. To achieve strong guarantees, a party issues transactions relative to an account. \(\Pi_{MT}\) models a secure channel between a sender and a receiver [2]. \(\Pi_{Sig}\) models a digital signature scheme and is parameterized by algorithms to generate the key, to sign, and to verify the signature [2]. In our instantiation, an honest participant of the network will issue only signed transactions on the network. Each party \(P\) has a signature key pair \((pk, sk)\).

\(^2\)An adversary is unable to replace the FP scanner with a malicious one.
obtained via $\mathcal{F}_{\text{Sig}}$. To submit an arbitrary transaction $tx'$, $P$ signs $tx'$ and appends the signature $\sigma$ and its party identifier $P$ to the transaction (i.e., the transaction format is $tx = (tx', \sigma, P)$). The verification key of the signature scheme can be extracted from the party identifier $P$. In the real world, $\mathcal{F}_{\text{Ledger}}$ can be realized with any blockchain system and a reasonable choice is Hyperledger Fabric [1]. The functionality $\mathcal{F}_{\text{SMT}}$ can be implemented by establishing a secure communication channel between parties (e.g., via TLS protocol). $\mathcal{F}_{\text{Sig}}$ can be accomplished by employing a standard signature scheme (e.g., ECDSA). In the real world, the functionality $\mathcal{F}_{\text{FP}}$ is realized by physical FP scanners that are able to verify the originality of a product. The list of all allocated devices is available to all parties. This can be achieved in the real world by involving an FP scanner issuer $D$. $D$ makes publicly available which FP scanner is associated with which party.

### 6.1 The Device Protocol $\Pi_{\text{FP}}$

The $\mathcal{F}_{\text{FP}}$ functionality can be implemented using a scanner provided by a device producer $D$. We rely on trust assumptions reported in Section 5.2 that typically require tamper-proof hardware. With our formalism, an honest party can create a hardware token implementing any desired efficient functionality and an adversary, given the token, can only observe the input/output of this token [14]. Typically (e.g., smart cards), tamper-proof hardware can also generate digital signatures to certify the outputs that it produces. This makes a FP scanner uniquely identifiable and capable of authenticating its outputs.

An FP scanner supports a training operation and a verification operation. The training permits the device to extract the fingerprint of a possessed good. Similarly, having a good and a fingerprint (i.e., the information representing a category of goods), the verification permits the device to check that the given good belongs to the category of goods associated with that fingerprint.

#### 6.1.1 $\Pi_{\text{FP}}$ Description

The operations performed by the $\Pi_{\text{FP}}$ protocol are stored on a ledger. Once the device is created, $D$ submits a corresponding transaction via $\mathcal{F}_{\text{Ledger}}$, to indicate that the device is created. To protect the validity of transactions, $D$ generates the signatures of the content of the transaction through $\mathcal{F}_{\text{Sig}}$. Once the device is created, $D$ handovers the device to a party $P_i$. To record this operation $D$ submits an appropriate transaction indicating that an FP scanner has been sent to $P_i$. $D$ generates a signature $\sigma$ through $\mathcal{F}_{\text{Sig}}$. Any party $P_i$ that owns the device can decide to handover the device to a party $P_j$. This operation is analogous to the previous handover operation. Any party $P_j$ that owns the device can perform training on a good to obtain the fingerprint $\text{catFP}$ of the category of the good. To record this operation, $P_j$ submits a transaction that stores the fingerprint of the good analyzed. $P_i$ generates $\sigma$ through $\mathcal{F}_{\text{Sig}}$. Any party $P_i$ that owns the device can perform a verification on a good given the expected category identifier. The verification output can be stored by submitting a specific transaction that reports the digital information $\text{auditData}$ returned by the device. If the device issuer $D$ that issues a device $\text{deviceId}$ needs to withdraw this device from the market (i.e., this device is broken), $D$ submits a specific transaction via $\mathcal{F}_{\text{Ledger}}$. This transaction indicates that the device cannot be used anymore. Every user can read the blockchain to obtain information about devices and asset categories stored. We assume that the creation and withdrawal of devices is always performed honestly and no device issuer misbehaves, and moreover the training process is performed honestly since the process is supervised by an honest authority.

### 6.2 The Protocol $\Pi_{\text{SC}}$

The building block of the protocol $\Pi_{\text{SC}}$ is an SC-smart contract $C_{\text{SC}}$, that manages the supply chain activities. Each asset in the system has a unique identifier that is associated with data on the asset in the supply chain. Access control policies must be defined to model real-world constraints where each party, according to its role in the supply chain, can perform different operations on an asset. As explained in Section 3, we assume the existence of a single trusted registration authority $\text{Reg}$ in charge of enrolling other participants in the system with long-term credentials. The identity of $\text{Reg}$ is publicly known.

When the system is initialized, $\text{Reg}$ generates and stores a signature key pair $(pk_{\text{Reg}}, sk_{\text{Reg}})$ via $\mathcal{F}_{\text{Sig}}$, stores an initially empty list $\text{L}$ of certified parties’ credentials, and finally stores $pk_{\text{Reg}}$ in the genesis block of the ledger. The smart contract and the access control policies realize our functionality $\mathcal{F}_{\text{SC}}$. They are initialized during the setup phase of the system along with blockchain global parameters. When activated by a transaction, $C_{\text{SC}}$ verifies that the transaction is originated by a legitimate party according to the defined access control policies and to the current state of the ledger. Participants send registration requests to the registration authority $\text{Reg}$ that after proper checks enrolls the participants with long-term credentials. Upon receiving the enrollment request from a party $P_i$ via $\mathcal{F}_{\text{SMT}}$, $\text{Reg}$ signs the identity of $P_i$ together with $P_i$’s public key $pk_i$ and attributes $\text{attr}_i$. The signature together with $pk_i$, $\text{attr}_i$ represents the $P_i$’s credential for the system. $\text{Reg}$ stores $P_i$’s identity credentials in $\text{L}$ and sends a confirmation message to $P_i$.

The enrollment procedure of $D$ consists of the generation of a signature key pair $(pk_{D}, sk_{D})$ via $\mathcal{F}_{\text{Sig}}$ and a unique identifier $D$ and sending of $pk_{D}$, $D$ and some arbitrary attributes $\text{attr}_D$ to $\text{Reg}$. If $\text{Reg}$ enrolls $D$, $D$ stores the tuple $(\text{register}, D, pk_{D}, \text{attr}_{D})$. Every time $D$ generates a new device calling $\mathcal{F}_{\text{FP}}$ with input $(\text{init}_D, D)$ a new device with identity $\text{deviceId}$ is created. To withdraw a device from the market (e.g., this specific device type is not working properly), $D$ calls $\mathcal{F}_{\text{FP}}$ with $\text{deleteDevice, deviceId}$. The FP scanner with identity $\text{deviceId}$ cannot be used anymore by parties in the system (i.e., every time an operation is done with a device, the system checks that this device is not withdrawn).

Let us now describe the protocol for a party $P_i$ not already analyzed before. At the setup, $P_i$ generates a key pair corresponding to its identifier and requests to be enrolled by contacting $\text{Reg}$. If the authentication is successful, $P_i$ receives the credentials $\sigma_i$ that uses to perform operations in the system. When $P_i$ wants to be provided with an FP scanner, $P_i$ informs $D$ with the message $(\text{device}, P_i)$ and $D$ sends message $(\text{handover}_D, D, P_i)$ to $\mathcal{F}_{\text{FP}}$. If party $P_i$ wants to create a new element, $P_i$ sends via $\mathcal{F}_{\text{Ledger}}$ a transaction with the unique identifier $\text{tokenId}$, the category $\text{categoryID}$, and the production field identity $\text{areaId}$ of the element. If $P_i$ wants to record a new product area, it submits via $\mathcal{F}_{\text{Ledger}}$ an appropriate transaction with the unique identifier $\text{areaId}$. If $P_i$ wants to create a new element, that is obtained by combining other stored items, it submits an
appropriate transaction containing the identifier item1d of the new elements, the list items of elements to combine, and the category categ1d to which the element belongs. If P_i performs a training on a category categ1d, P_i sends (train_{FP}, categ1d) to F_{FP}. If the training returns catFP, P_i submits an appropriate transaction containing categ1d and catFP. Every time P_i inspects an item using an allocated FP scanner, P_i verifies the item sending (eval_{FP}, P_i, item1d, categ1d) to F_{FP}. If the verification has returned a result (result, device1d, item1d, categ1d), P_i submits an appropriate transaction containing the identity asset1d of the two parties and the asset to transfer. If P_i wants to disaggregate a batch to retrieve stored items, it submits an appropriate transaction containing the identifier batch1d of the batch and the list product of products to aggregate. If P_i wants to disaggregate a batch to retrieve stored items, it submits an appropriate transaction containing the batch identifier batch1d. If P_i wants to update an asset’s state, it submits an appropriate transaction containing the new state and the asset identifier. This operation can only set the state from intact or packaged to destroyed. If P_i wants to transfer an asset to another party P_j, P_i submits an appropriate transaction containing the identifier of the two parties and the asset to transfer. P_j can decide to accept or reject the asset by submitting an appropriate transaction reporting the decision.

When P_i wants to retrieve an asset state of an asset with identifier asset1d, it relies upon blockchain consensus nodes, which process transactions.

The full version of our work [4] provides the proof of the following theorem.

**Theorem 6.1.** For any PPT adversaries A, there exists a PPT simulator S such that for all non-uniform PPT distinguisher Z, the protocol Π_{SC} UC-securely computes F_{SC} in the (F_{FP}, F_{Ledger}, F_{SMT}, F_{Sig})-hybrid model.

### 6.3 Improvements and Optimizations

- It is possible to decentralize the registration authority and the FP scanner issuer. Indeed, the enrollment of parties can be performed using threshold signatures to sign messages under the guarantee that sufficiently many entities realizing the registration authority behave honestly. Secret sharing can be used to share the secrets needed by the parties realizing the decentralized registration authority. Using the same tools we can also relax the trust assumption made on the FP scanner issuer in Section 3 via decentralization. Notice that if we have several authorized FP scanner issuers, even if an FP scanner issuer misbehaves by creating malicious FP scanners, this behavior can be detected by other honest parties equipped with trusted FP scanners provided by an honest FP scanner issuer.

- In some cases, data must be stored on-chain for audit purposes but on the other side this might affect data confidentiality. For example, a wholesaler might want to hide the number of products purchased from each distributor or a manufacturer might want to hide the number of raw materials used per product. To avoid counterfeiting, the product tracing system must guarantee that a party cannot use more units of a product than what it legitimately possesses. Solutions such as homomorphic encryption, threshold decryption, commitment schemes and zero-knowledge proofs can preserve confidentiality, public verifiability, and correctness of data in blockchains. In homomorphic encryption, the encryption function allows combining two ciphertexts into a third ciphertext encoding a plaintext that is related to the original two plaintexts. Threshold decryption allows parties to decide to decrypt just a specific ciphertext that can be the one obtained through homomorphic operations. A commitment scheme, instead, allows a party to commit to a secret value while keeping it hidden from others, with the ability to reveal the committed value later. It is infeasible for the committer to change the value after it has been committed. Confidential data can be stored off-chain and next it is possible to prove properties about such data through zero-knowledge proofs. This allows to preserve confidentiality while at the same time data integrity remains publicly verifiable.

- During the enrollment of a party, Reg can establish some additional attributes that can be used also to add further constraints on the operations that a party can perform according to the role. In the case of Producers or Manufacturers, they must be able to create new products only belonging to the categories that have been approved by Reg during the registration.

### 7 CONCLUSIONS

In this work, we have proposed and analyzed a system for tracing physical goods leveraging blockchain technology. As already presented in previous works we exploit blockchain to design a general model for blockchain-based supply chains. In contrast to prior work, we have explicitly modelled the management of digital twins within the traceability and the verification of goods. Our protocol Π_{FP} is secure according to the paradigm of secure computation and can be implemented using Hyperledger Fabric [1] as blockchain, which can support, under some specific settings, about 20 thousand transactions per second [9]. One can therefore think of using our system for the protection of high-value goods. There normally are quite limited amounts of such goods, and thus the corresponding transactions reaming within the limits of the employed technologies. A core component of our system is the possibility of verifying the correspondence among physical goods and their on-chain digital twins. Moreover, our system is based on a model that can be easily adapted to accommodate additional features (e.g., secret business).

**ACKNOWLEDGMENTS**

We thank Giorgio Ciardella for several remarks and explanations on real-world food traceability and the effectiveness of biological fingerprint scanning. This work has been supported in part by a scholarship funded by Farzati Tech, in part by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIVILEDGE), in part by a scholarship funded by Campania Region, and in part by the
European Research Council (ERC) under agreement No 885666 (project PROCONTRA).

REFERENCES

[1] Eli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Chrisidis, Angelo De Caro, David Eneyart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Murudharian, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vuokko, Sharon Weed Cocco, and Jason Yelick. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In EuroSys 2018. ACM, 30:1–30:15. https://doi.org/10.1145/3190508.3190538

[2] Eli Androulaki, Jan Camenisch, Angelo De Caro, Maria Duhovniskaya, Kaooster Elkhayasui, and Bjorn Tackmann. 2020. Privacy-preserving auditable token payments in a permissioned blockchain system. In AFT ’20 ACM, 255–267. https://doi.org/10.1145/3419614.3423259

[3] Tomasso Aste, Paolo Tasca, and Tiziana di Matteo. 2017. Blockchain Technologies: The Foreseeable Impact on Society and Industry. Computer 50, 9 (2017), 18–28. https://doi.org/10.1109/MC.2017.3571664

[4] Vincenzo Botta, Laura Fusco, Attilio Mondelli, and Ivan Visconti. 2021. Secure Blockchain-Based Supply Chain Management with Verifiable Digital Twins. CoRR abs/2109.03570 (2021).

[5] Ran Canetti. 2001. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In FOCS 2001. IEEE Computer Society, 136–145. https://doi.org/10.1109/SFCS.2001.959888

[6] Thijs Defrants, Chandrima Shrivastava, Tarl Berry, Pieter Verboven, Daniel Onwude, Seraina Schudel, Andreas Buhlmann, Paul Cronje, and Rene M. Rossi. 2021. Digital twins are coming: Will we need them in supply chains of fresh horticultural produce? Trends in Food Science & Technology 109 (2021), 245–258. https://doi.org/10.1016/j.tifs.2021.01.025

[7] Tianhu Deng, Keren Zhang, and Zuo-Jun (Max) Shen. 2021. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. Journal of Management Science and Engineering 6, 2 (2021), 125–134. https://doi.org/10.1109/jmse.2021.00303

[8] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1987. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In STOC 1987. ACM, 218–229. https://doi.org/10.1145/28395.284240

[9] Christian Gorstello, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. 2020. FastFabric: Scaling hyperledger fabric to 20,000 transactions per second. Int. J. Netw. Manag. 30, 5 (2020). https://doi.org/10.1002/ijnm.2099

[10] Michael Grieves. 2014. Digital twin: manufacturing excellence through virtual factory replication. White paper 1 (2014), 1–7. https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/DELMA/PRODUCTS-SERVICES/DELMA-APRISO-Digital-Twin-Whitepaper.pdf

[11] Daqing Guo, Shiquan Ling, Hao Li, Di Ao, Tongda Zhang, Yiming Rong, and George Q. Huang. 2020. A framework for personalized production based on digital twin, blockchain and additive manufacturing in the context of Industry 4.0. In CASE 2020. IEEE, 1181–1186. https://doi.org/10.1109/CASE48305.2020.9216732

[12] Sohail Jabbar, Huw Lloyd, Mohammad Hammadouh, Balamide Adesib, and Umar Raza. 2021. Blockchain-enabled supply chain: analysis, challenges, and future directions. Multimedia Systems 27, 4 (2021), 787–806. https://doi.org/10.1007/s00530-020-00687-0

[13] David Jones, Chris Snider, Aydin Nasoohi, Jason Yon, and Ben Hicks. 2020. Characterizing the Digital Twin: A systematic literature review. CRP Journal of Managing Manufacturing Science and Technology 29 (2020), 36–52. https://doi.org/10.1016/j.jcrpj.2020.02.002

[14] Jonathan Katz. 2007. Universally Composable Multi-party Computation Using Tamper-Proof Hardware. In EUROCRYPT 2007 (LNCS, Vol. 4515). Springer, 115–128. https://doi.org/10.1007/978-3-540-72540-4_7

[15] Dongjin Lee, Sang Hyun Lee, Neda Masoud, M.S. Krishnan, and Victor C. Li. 2021. Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. Automation in Construction 127 (2021), 103688. https://doi.org/10.1016/j.autcon.2021.103688

[16] Jiongbin Liu, William Yeoh, Youyang Qu, and Longxuan Gao. 2022. Blockchain-based Digital Twin for Supply Chain Management: A Literature Review and Future Research Directions. CoRR (2022). https://arxiv.org/abs/2202.03966

[17] Tsega Y. Melesse, Valentina Di Pasquale, and Stefano Riemma. 2021. Digital Twin-Whitepaper.pdf

[18] Luis F. Rivera, Hauk A. Müller, Norha M. Villegas, Gabriel Tamura, and Miguel Jiménez. 2020. On the Engineering of IoT-Intensive Digital Twin Software Systems. In ICSE ’20. ACM, 631–638. https://doi.org/10.1145/3379748.3392195

[19] Jose Antonio Marmolejo Saucedo, Margarita Hurtado-Hernández, and Ricardo Suarez-Valdes. 2019. Digital Twins in Supply Chain Management: A Brief Literature Review. In IC3. 653–661. https://doi.org/10.1007/978-3-030-33585-4_63

[20] Letizia Tebaldi, Giuseppe Vignali, and Eleonora Bottani. 2021. Digital Twin in the Agri-Food Supply Chain: A Literature Review. In AFMS 2021 (IFIP, Vol. 633). Springer, 276–283. https://doi.org/10.1007/978-3-030-85910-7_29

[21] Feng Tian. 2016. An agri-food supply chain traceability system for China based on RFID blockchain technology. In ICSISM 2016. 1–6. https://doi.org/10.1109/ICSSISM.2016.7538424

[22] C. N. Verdouw, H. Sundmaeker, F. Meyer, J. Wolpert, and J. Verhoeven. 2013. Smart Agri-Food Logistcs: Requirements for the Future Internet. In Dynamics in Logistics (Lecture Notes in Logistics). Springer, 247–257. https://doi.org/10.1007/978-3-642-35966-8_20

[23] Birz Yaqob, Khaled Salah, Maseen Uddin, Raja Jaryaram, Mohammed A. Omar, and Muhammad Imran. 2020. Blockchain for Digital Twins: Recent Advances and Future Research Challenges. IEEE Network 34, 5 (2020), 290–298. https://doi.org/10.1109/MNET.001.1900861