symmetry

MDPI

Article

An Anti-Counterfeit and Traceable Management System for Brand Clothing with Hyperledger Fabric Framework

Chin-Ling Chen [1,2,3,*], Xin Shang [1,*], Woei-Jiunn Tsaur [4], Wei Weng [1,*], Yong-Yuan Deng [3,*], Chih-Ming Wu [5,*] and Jianfeng Cui [6]

1 School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
2 School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
3 Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
4 Computer Center, National Taipei University, New Taipei City 237303, Taiwan; wjtsaur@mail.ntpu.edu.tw
5 School of Civil Engineering and Architecture, Xiamen University of Technology, Xiamen 361024, China
6 School of Software Engineering, Xiamen University of Technology, Xiamen 361024, China; jfcui@xmut.edu.cn
* Correspondence: clc@mail.cyut.edu.tw (C.-L.C.); 2022031426@s.xmut.edu.cn (X.S.); wwweng@xmut.edu.cn (W.W.); allendeng@cyut.edu.tw (Y.-Y.D.); chihmingwu@xmut.edu.cn (C.-M.W.)

Abstract: Counterfeit products are internationally regarded as "the world's second greatest public health hazards after drugs". Counterfeiters produce counterfeit brand clothing and then sell them to consumers through unofficial channels; thus, consumers spend a lot of money without getting the value they deserve. With the rise of e-shopping, the safety and security of branded clothing supply chains are also under threat. Counterfeit branded apparel manufacturers generate profits while genuine manufacturers suffer, which ultimately violates the interests of the public. This study proposes a traceable anti-counterfeit management system for branded clothing based on Hyperledger Fabric technology. This system can achieve full traceability of the production information of branded clothing. It uses the unique characteristics of blockchain, such as being unforgeable, traceable, open, and transparent, and collectively 'maintaining', to record the specific production processes of the brand clothing, and ensure the authenticity and legitimacy of the production information of brand clothing. The end-user can self-verify the product's authenticity by sharing the product's details on the immutable framework. It solves problems surrounding information asymmetry, opaque supply chain data, and easy falsification in the production process of branded clothing in traditional branded clothing supply chains.

Keywords: blockchain; internet of things technology (IoT); anti-counterfeit traceability; ECDSA; Hyperledger Fabric; clothing supply chain

Citation: Chen, C.-L.; Shang, X.; Tsaur, W.-J.; Weng, W.; Deng, Y.-Y.; Wu, C.-M.; Cui, J. An Anti-Counterfeit and Traceable Management System for Brand Clothing with Hyperledger Fabric Framework. *Symmetry* **2021**, *13*, 2048. https://doi.org/10.3390/sym13112048

Academic Editor: José Carlos R. Alcantud

Received: 26 September 2021
Accepted: 22 October 2021
Published: 31 October 2021

1. Introduction

1.1. Background

Branded clothing is favored by consumers, and is a target for counterfeiters who profit illegally. Regarding a brand-name product, especially an international brand, one must focus on the quality of the product, as well as the production, management, and use of legal methods to defend the interests and honor of the brand against counterfeiting. The situation is global, with up to 5% of EU imports being counterfeit [1]. Counterfeiting rings not only operate formulaically, to sell counterfeit products, they also exploit the transmission effects of the internet, to dump counterfeit goods through informal channels. Although social media networks are not the primary venues for commerce, such platforms are powerful tools for marketing products. They can drive sales to online stores with little or no regulation. The most extensive online platforms—YouTube, Facebook, WhatsApp, WeChat, Instagram, and TikTok—have billions of active users, and just as many potential customers. Studies have shown that almost one-fifth of the content posted on social media

concerning branded apparel is illegal [2]. According to statistics, in 2020, the fashion industry will lose more than USD 50 billion due to counterfeit products. Designer clothing are the most counterfeited products, followed by cosmetics, watches, jewelry, and luggage. By 2022, the global trade in fakes is expected to reach USD 4.2 trillion [3].

From the above data, it is clear that counterfeiters make huge illegal financial gains by falsifying branded clothing, to the detriment of consumers. Moreover, cross-border issues deserve attention. Many counterfeiters sell foreign branded clothing in their own countries. Because customers do not have enough understanding of foreign brands, it is easy to buy counterfeit clothes, which could lead to certain economic losses. The root cause of this phenomenon is that consumers are unaware of the production process of branded clothing in the traditional supply chain. The entire process, from design, production, to sales of branded clothing, goes through many links, and any problems will eventually be harmful to consumers. In the traditional supply chain, only the brand company, manufacturer, and retailer have the core information of the brand clothing. The end consumer does not know the whole supply process of the brand clothing. This results in opaque and unequal information in the supply chain. In this model, consumers can easily suffer financial losses due to information asymmetry. Therefore, effective management methods and traceability technologies are the only way to achieve information symmetry between buyers and sellers.

As for the centralized management of traditional blockchain—we used blockchain technology to conduct decentralized control of data in the supply chain. Blockchain technology can solve the information opacity in brand clothing supply chains, protect the end consumers, and defend the legitimate rights of anti-counterfeiting detection. Therefore, this study proposes an anti-counterfeiting management system for traceable branded clothing based on Hyperledger Fabric technology. Most new technologies rely on the futuristic characteristics of the internet [4]. We utilized a decentralized, open, autonomous, and immutable blockchain for anti-counterfeit traceability of branded clothing through the effective combination of internet of things technology and blockchain technology. Hyperledger Fabric, as a new blockchain framework, focuses more on privacy protection and performance expansion and performs better than Ethereum in large supply chain operations. Therefore, Hyperledger Fabric technology is proposed as the main technology for research and discussion. More attention will be paid to data security, privacy protection, and regulations from the clothing supply chain to the commercial supply chain. Blockchain technology can provide good support in regard to the needs of the business supply chain. The combination of blockchain and supply chain management could solve problems and challenges, including the development trends surrounding commercial supply chains in the future. Therefore, blockchain-based product traceability systems are receiving increasing attention from the industry and academia [5].

1.2. Related Works

Traditional anti-counterfeiting methods on the market, regarding branded clothing, include trademark anti-counterfeiting, logo anti-counterfeiting, wheat washing anti-counterfeiting, laser engraved buttons, semi-invisible pattern lining, anti-counterfeit sewing thread, and direct printing anti-counterfeit marks on garments. However, these current anti-counterfeiting methods have some shortcomings. For example, it is difficult for consumers to recognize and identify (authenticity), and anti-counterfeiting costs are relatively high [6]. For this reason, many ideas have been proposed to get rid of these traditional (and easily forged) anti-counterfeiting methods, e.g., by combining modern technology with traditional techniques. Table 1 presents different anti-counterfeiting solutions for clothing. Although some articles use blockchain technology, they still have some shortcomings.

Table 1. Comparison with existing anti-counterfeiting traceability methods.

Authors	Year	Objective	Technologies	Merits	Demerits
Alzahrani et al. [7]	2018	The combination of blockchain technology and NFC technology in the internet of things is used to prevent counterfeiting of fake product technologies.	Blockchain and consensus protocol.	Blockchain technology is used to solve the disadvantages brought by centralized management and to crack down on fake and shoddy products.	No specific data flow frameworks are proposed.
Zhu et al. [8]	2020	Blockchain technology is used for anti-counterfeiting traceability of the drug supply chain.	Blockchain and RFID.	Ensure the integrity of drug information management and a high level of privacy protection.	Encryption and decryption methods need improvement.
Bullón Pérez et al. [9]	2020	Ensure the transparency of the supply chain, the authenticity, reliability, and integrity of clothing, and the effectiveness of the retail end product.	Blockchain and hash functions.	Use a private and open blockchain to track products. Blockchain participants are proposed for each production stage.	The specific process of clothing production is not put forward.
Yin et al. [10]	2021	An NFC-enabled anti-counterfeiting system (NAS) is proposed.	Blockchain and near-field communication.	A secure and immutable scientific data provenance tracking and management platform with provenance records.	Lack of connection with actual cases for discussion.
Agrawal et al. [11]	2021	Investigates and proposes a blockchain-based traceability framework for traceability in the multitier textile and clothing supply chain.	Blockchain and smart contract.	The internet of things technology and blockchain technology are combined to carry out anti-counterfeiting traceability of clothing.	The specific flow of data are not reflected.

Alzahrani et al. [7] similarly expressed that traditional supply chains have a single point of processing, storage, and failure problems in regard to anti-counterfeiting traceability through a centralized authority. Therefore, the technology of 'block supply chain' is proposed, which is a new decentralized supply chain that uses blockchain and communication technology to detect counterfeit attacks. However, the processes of some data are not well expressed. The framework proposed by Zhu et al. [8] for anti-counterfeit traceability of pharmaceuticals based on blockchain is also worth learning, which can ensure the transparency and openness of the pharmaceutical supply chain; the smart contract-based access control policy model is about preventing the drug information from being changed or disclosed at the nodes of the blockchain. The security of the framework would go further if the privacy of the data were enhanced. Bullón Pérez [9] present an updated traceability scheme and proposal for the apparel industry for ready-to-wear apparel, tracking suppliers and customers throughout the logistics chain. However, less specific data flow frameworks make this aspect unconvincing. Yiu et al. [10] present a feasible mechanism for developing a product orientation and traceability ecosystem using blockchain technology, mainly through a series of security and threat analyses, mainly for Near Field Communication (NFC) enabled anti-counterfeiting systems to identification. There is a lack of discussion in conjunction with physical objects. Agrawal et al. [11] presented a specific case study using blockchain technology to verify and track the supply chain of off-the-shelf apparel. However, it is difficult for buyers to self-verify the transaction information on the blockchain, as they lack some IoT technology.

The methods proposed above are all dedicated toward product anti-counterfeit traceability, but the current anti-counterfeit traceability methods still have some problems for buyers and supply chain participants. For buyers, currently proposed anti-counterfeit traceability methods do not provide a platform for anti-counterfeit inquiries and an arbitration mechanism. For supply chain participants, the current problem is that it is difficult to effectively manage the supply chain, find the source of the problem, and there is the lack of an arbitration mechanism to maintain the supply chain. In this paper, we propose an anti-counterfeit and traceable management system for brand clothing with the Hyperledger Fabric framework. This paper involves a cryptographic mechanism to encrypt the data, which further ensure the security of the data. It also combines the internet of things and blockchain technologies to trace the supply chain information of the production and sales process of branded clothing in real-time. The characteristics of blockchain are used to ensure the transparency and traceability of data in the supply chain. The proposed scheme ensures the correct transmission of data.

The paper is written in the following structure. Section 2 focuses on the relevant techniques used in our proposed scheme. Section 3 presents our specific proposal and the detailed process. In Section 4, we perform a security analysis and a discussion of the relevant features of the scheme. Section 5 gives a discussion of computational costs, communication performance, and comparisons. Finally, in Section 6, we present a summary of our scheme.

2. Preliminary

2.1. Blockchain

Blockchain is an important concept of Bitcoin, which is essentially a distributed database [12]. It has the characteristics of high reliability and high confidentiality and has good prospects in regard to effectively solving the trust problem between the two parties [13]. As the underlying technology of Bitcoin, blockchain is a string of data blocks generated using cryptographic methods of correlation, each containing information about a batch of Bitcoin network transactions, used to verify the validity of its information and to generate the next block. Blockchain technology has the following five characteristics: (1) characteristic distributed database; (2) uniqueness—each record is a timestamp and cannot be tampered; (3) transparency of data; (4) irreversibility of records; and (5) traceability. The application of blockchain technology in a supply chain and logistics is widely recognized as it records and stores all transaction information of stakeholders in the supply chain in a tamper-evident manner. It makes supply chain information more transparent and the receipt of information more symmetrical. Blockchain may significantly impact supply chain management, its relationships, and governance structures [14].

2.2. Hyperledger Fabric

Hyperledger Fabric (Hyperledger Architecture) is an open-source project launched by the Linux Foundation in 2015 to advance blockchain digital technology and transaction validation, with investments in industry giants, including finance, manufacturing, logistics shipping, and security consulting [15]. Hyperledger Fabric builds on the foundation of public chains to create an efficient, low-cost operating model, known as the federated chain model. Areas of application include, but are not limited to, dispute resolution, trade logistics, foreign exchange netting, food safety, contract management, diamond provenance, reward point management, low liquidity securities trading and settlements, identity management, and settlements via digital currencies [16]. Compared to the previous Ethereum platform, Hyperledger Fabric addresses performance scalability and privacy issues through fine-grained access control. Compared with Hyperledger Sawtooth, Hyperledger Fabric establishes the concept of the channel, which provides more comprehensive data protection and can better resist attacks from attackers. It is clear that Hyperledger Fabric allows fine-grained control over consistency, which improves performance, scalability, and privacy [17]. The introduction of the channel in Hyperledger Fabric plays a very good role in

data privacy protection [18]. One of the frameworks of Hyperledger Fabric is shown in Figure 1.

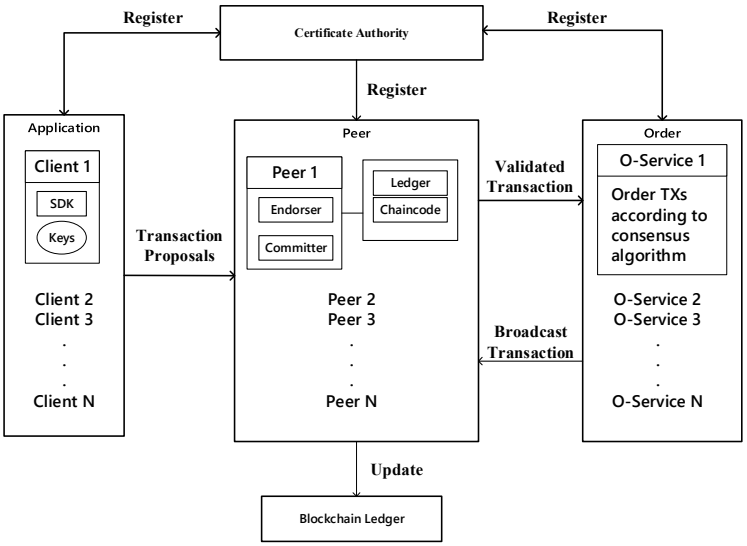

Figure 1. Hyperledger Fabric framework diagram.

From Figure 1, we can see that the Hyperledger Fabric framework consists of four main parts: the application, the peer node, the order (ordering service node), and the certificate authority (CA) node.

Application: the Hyperledger Fabric provides an APP platform for clients to allow people or departments involved in the supply chain to more easily interface to the transactions within the blockchain. The initiation of transactions is done through the Software Development Kit (SDK), the SDK version is Fabric Node SDK 1.4. All communication messages need to contain signatures, and the signature certificates and public and private keys are obtained through the CA node. The client submits a transaction proposal to the endorsing node (endorser) and obtains the endorsed transactions from the endorser node. After collecting enough transactions, it broadcasts them to the sorting service node.

Peer node: the peer node is the main body involved in the transaction, which can represent each member involved in the chain, and is responsible for the execution of the smart contract in the consensus link. It contains various types of nodes; the main participating transaction nodes are the endorser node and committer node. The peer node also stores the ledger data and the chain code. After receiving a transaction proposal from the client, the endorsing node verifies the transaction signature, simulates the transaction's execution, performs a signature endorsement on the result, and then sends the validated transaction to the order. The endorsing nodes are dynamic roles and are bound to specific chain codes that specify which nodes must complete a valid transaction endorsement. The peer node is the endorsing node only when the client initiates a transaction endorsement request to the peer node; otherwise, the peer node is just an ordinary committer node. It is only responsible for verifying the transaction.

Order: the primary function is to sort the transactions to ensure data consistency on each peer node. Order sorts transactions according to consensus algorithms and broadcasts the sorted transactions into the blockchain. The peer node updates the transactions to the ledger after passing the transaction information broadcasts by the order node to ensure data integrity. After receiving the sorted transaction information of the order node, the peer node will upload its data to the blockchain network and update the data in the blockchain network.

CA node: the peer node is responsible for authorizing and authenticating all nodes that join the blockchain, including the upper layer clients. Each of them has its certificate issued for Identification in the transaction process. For the Hyperledger Fabric blockchain

framework, there are high requirements for authentication, from which, CA nodes play the role of authorizing users and signing transmissions. Peer nodes use digital certificates issued by CA for authentication, encrypting data transmission, authorizing users, managing user certificates, and other features to ensure data security in the blockchain network.

2.3. Elliptic Curve Digital Signature Algorithm (ECDSA)

Scott Vanstone first proposed the elliptic curve signature algorithm (ECDSA) in 1992 [19]. The security of the elliptic curve crypto signature regime is based on the intractability of the elliptic curve discrete logarithm problem. The elliptic curve discrete logarithm problem is far more complex than the discrete logarithm problem, and elliptic curve cryptosystems have higher cryptographic strength. This brings the benefits of smaller computational parameters, shorter keys, faster operations, and shorter signatures. Elliptic curve ciphers are therefore particularly suitable for applications where processing power, storage space, bandwidth, and power consumption are limited.

We can analyze how ECDSA works by simulating the signing and verification of messages M between user A and user B using ECDSA. User A first sets the elliptic curve parameters $y^2 = (x^3 + ax + b) \bmod p$, its corresponding key pair (d_A, Q_A), d_A is the private key of user A, Q_A is the public key of user A. User A can compute the public key from the private key $Q_A = d_A G$, where G is the parameter point in the ECDSA elliptic curve.

Signature process:

1. User A generates a random number k based on the ECDSA algorithm, uses points G to calculate the public key $(x, y) = kG$.
2. User A calculates the hash value of the message M: $h = H(M)$.
3. User A calculates the eigenvalues of an elliptic curve (r, s), of which $r = x \bmod n$, $s = k^{-1}(h + rd) \bmod n$ (k^{-1} is the multiplicative inverse of the modulus of k).
4. (r, s) is the digital signature of user A. User A sends the elliptic curve parameters $D = (p, a, b, G, n, h)$ and User A's public key Q_A to User B for verification of the correctness of the signature.

Validation process:

1. User B calculates the hash value of the message M: h'.
2. Calculation $u_1 = h's^{-1} \bmod n$ and $u_2 = rs^{-1} \bmod n$.
3. The calculation $(x', y') = u_1 G + u_2 Q_A$, if $x' = r \bmod n$ then the signature is validated.

2.4. Smart Contract

Blockchain is a global decentralized distributed database ledger, and smart contracts are software programs executed in a decentralized manner based on blockchain technology [20]. Blockchain technology can achieve collaboration and trust between multiple enterprise entities through smart contracts, thereby expanding the scope and depth of mutual cooperation between parties [21]. An event-driven, stateful program runs on top of the blockchain system and can hold and process digital assets on the blockchain ledger. The development of blockchain technology provides an excellent operating basis for smart contracts, which can play an important role in the blockchain. Smart contracts guarantee important features of blockchain technology: data cannot be deleted or modified, only added, ensuring traceability of history, while the cost of doing malicious acts will be high, as its malicious acts will be recorded forever.

Smart contracts based on blockchain technology can bring into play the advantages of smart contracts, in terms of cost efficiency, and avoid the interference of malicious acts in the normal execution of contracts. The smart contract is written into the blockchain in digital form. The characteristics of blockchain technology guarantee that the whole process of storage, reading, and execution is transparent, traceable, and unchangeable. At the same time, a state machine system is built by the consensus algorithm that comes with the blockchain, which enables smart contracts to operate efficiently.

2.5. BAN Logic

BAN logic, introduced by Burrows, Abadi, and Needham, is a cognitive logic for analyzing security protocols. It models the knowledge of topics in a protocol at a level of abstraction [22]. Specifically, BAN logic helps users determine whether the information exchanged is trustworthy, resistant to eavesdropping, or both. BAN logic begins in situations where information is vulnerable to tampering during the exchange period. Typical BAN logic includes: (1) verifying the source of the message; (2) verifying the 'freshness' of the message; (3) verifying the credibility of the source.

2.6. Threat Model

In related works, we conducted a brief review of previous research and identified problems and shortcomings that some research currently has. As a result, we compiled some possible threat patterns usually caused by security issues and vulnerabilities in the system. The security of the blockchain is also relative and, therefore, this system is potentially risky.

1. Mutual authentication: the current internet environment is due to the network nature of blockchain. Participants need to authenticate with each other to determine whether they are receiving correct and secure messages. This feature comes about precisely because of the decentralized blockchain network. This also allows the communication between the two in the blockchain network to be secure and guaranteed.
2. Data integrity: when using the system for transactions, all data present in the system must be processed in an integrated manner. The system must ensure the integrity of the data when conducting transactions and the system must ensure that the data cannot be tampered with by anyone when it is transmitted and stored.
3. Non-repudiation: the transactions in the blockchain network are transmitted and stored through the chain code; therefore, to perform anti-counterfeiting and traceability verification, all transactions must be verified by an ECDSA digital signature to ensure the non-repudiation of the data. The deployment of the chain code ensures that the data have a non-repudiation effect and it makes the information more secure.
4. Known attack problems: blockchain networks are also subject to attacks by illegal nodes and illegal persons, which contain different types of attacks.
    (1). Man-in-the-middle attacks: in a blockchain network, when the sender communicates with the receiver, an attacker performs a man-in-the-middle attack by intercepting messages during the communication process. The specific process is: the attacker intercepts the message sent by the sender to the receiver and tampers with the message content. The attacker sends the altered message to the receiver instead of the sender to perform the man-in-the-middle attack.
    (2). Replay attack: an attacker will resend the data obtained after eavesdropping on the sender to the receiver untouched. The attacker does not need to know the exact meaning of the data, but only know what the data do to attack the receiver, by sending the data again without knowing its content.

3. Proposed Scheme

3.1. System Architecture

For the process of clothing production, from the time a company decides to start producing clothing to the time the finished garment appears. It goes through the following stages. The supply chain flow of brand clothing is shown in Figure 2—brand company (BC), material supplier (MS), manufacturer (MF), retailer (R) form the alliance chain. The alliance chain, together with the customer (CU) and third party (TP), will form a Hyperledger Fabric-based brand clothing anti-counterfeiting management system. The system framework is shown in Figure 2.

1. Brand company (BC): the brand clothing company. This role manages and determines the design and production of clothing. The production of brand clothing requires a

license from the company. The supply chain only works when the brand clothing company decides to produce the clothing.

2. Material supplier (MS): this role provides the raw materials for the production of clothing products. The BC will first send the production license of the clothing and the transaction information of all types of orders to MS, which will provide the raw materials according to the orders and keep a record of the origin of the raw materials.
3. Manufacturer (MF): first, the MF receives the raw materials and order information from the MS. The MF is then responsible for processing the raw materials into garments. MF will inlay the unique identification code on each garment. Each piece of clothing corresponds to a unique identification code. Domestic and foreign customers can scan the identification code through the client interface to enter the blockchain network to query the information of the brand clothing.
4. Retailer (R): the retailer will receive the garments from the manufacturer, confirm the order information, return it to the MF, and sell the garments to the customer, and then upload the transaction information to the blockchain.
5. Customer (CU): the customer can use the mobile app or scan the code to see all of the information about the product. Moreover, the CU could inspect the entire production process of a finished product.
6. Blockchain center (BCC): a Hyperledger Fabric-based blockchain framework that detects the legitimacy of participants. Moreover, it will record various information uploaded to the blockchain by participants.
7. Third Party (TP): all transactions can be accessed by linking to the blockchain network, regardless of whether the clothing is at the design stage, the raw material supply stage, the clothing design stage, or the retail stage. All information can be accessed to detect counterfeit or illegal branded products.

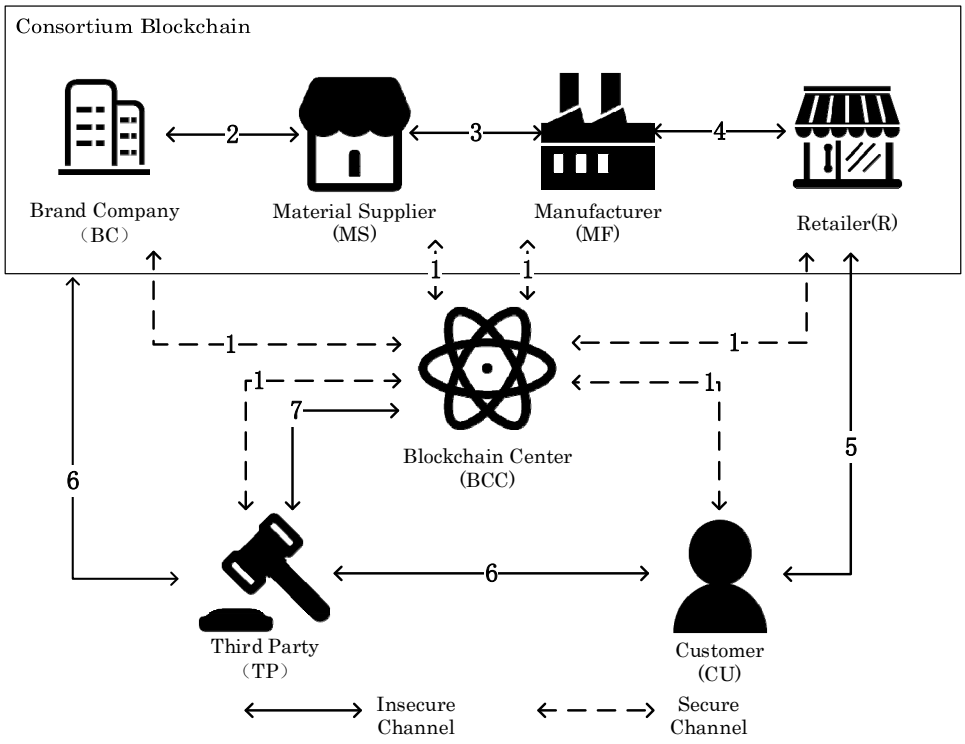

Figure 2. System framework diagram.

There are the seven 'actors' from the supply chain flow framework for a Hyperledger Fabric-based anti-counterfeit management system for traceable brand clothing. The specific process is shown in Sequence Figure 3. The specific steps are:

Step 1. This step is the registration phase for each role in the system. All participants in the clothing brand company (BC), material supplier (MS), manufacturer (MF), retailer (R), customer (CU), and third party (TP) need to be authenticated at the CA node in the blockchain network. After passing authentication, the roles can exchange information through the channel.

Step 2. When the BC wants to produce clothes, the BC first sends the production certificate, raw material order, and the production plan to the MS. The MS confirms the production information and starts to collect the raw material, and the MS sends a response message to BC when it is finished. BC uploads the relevant information to the blockchain center through the sorting node and updates the local ledger after ensuring the ordering information is correct.

Step 3. When the MS prepares raw materials for garment production, it will send the raw materials and production information to the MF, which will compare the production information with the raw materials received in reality to ensure the legitimacy of the data, and send a response message to the MS. When the MF processes the raw materials into garments, it will mark a unique identification code on each garment, and the CU can check the information of the garments on the blockchain, according to the identification code.

Step 4. The R will send the order to the MF, which will confirm the order from the R, provide the garment based on the order information, upload the confirmed order information to the blockchain center, and then send the garment and certificate to the R. After confirming that the garment information is the same as the order, the R will return the confirmed information to MF.

Step 5. The CU can purchase brand clothing through the R. After the R determines the transaction information applied by the CU, it will upload the transaction information to the blockchain center. At this point, the whole supply chain ends, and all data will be operated on the chain at various stages, and the integrity of the data will be guaranteed.

Step 6. The CU can use the identification code on the garment to join the blockchain network with an app or client to check the garment's production process and raw material information. If the CU finds out that he/she has purchased a counterfeit garment, he/she can submit an arbitration application with a third party.

Step 7. The third party will go to the blockchain center to examine the information based on the arbitration request provided by the CU for each transaction and operation in the supply chain, and check whether the name-brand clothing is correct or not in the supply chain during the process. The legitimacy and validity of the brand clothing can be verified by examining the signature information of each department.

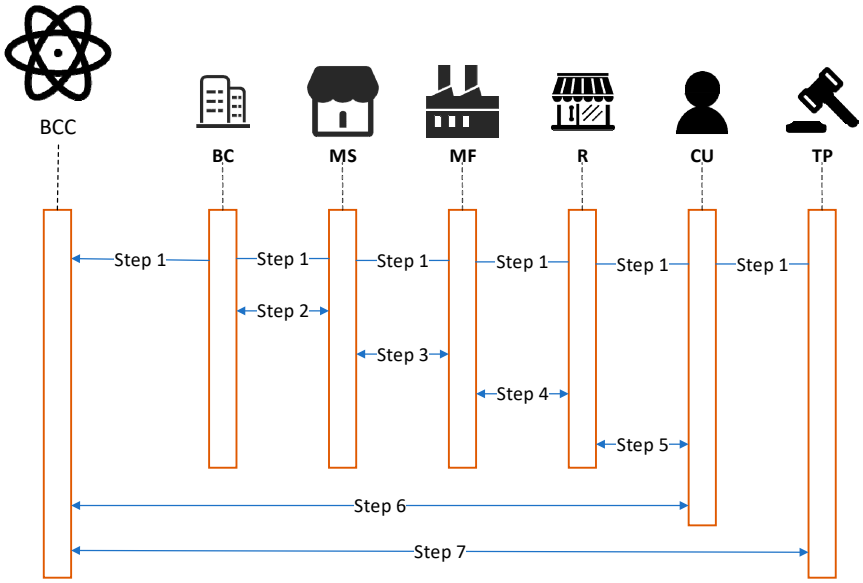

Figure 3. Stakeholders' transaction sequence diagram.

3.2. Hyperledger Fabric Detailed Transaction Information Flow

The transaction process in the Hyperledger Fabric-based blockchain framework is shown in Figure 4, which shows in detail how two companies in the supply chain interact with each other. If Company A and Company B interact for necessary information in the supply chain, Company A and Company B will each form an organization (as Organization A and Organization B). They will also register through a certificate authority (CA) node, which will return their corresponding public and private keys and digital identity credentials. Company A and Company B then create their respective Organization (Org), which will contain various nodes (endorser node, committer node, anchor node, and leader node), collectively known as peer nodes. Each peer node in an organization can deploy one or more chain codes and will store a copy of the ledger in the channel. The order node sorts and packages the transactions and information interactions generated in the channel, and uploads transaction information to the Blockchain Center. At this point, Company A and Company B can conduct transactions and information interactions in the Hyperledger Fabric network framework.

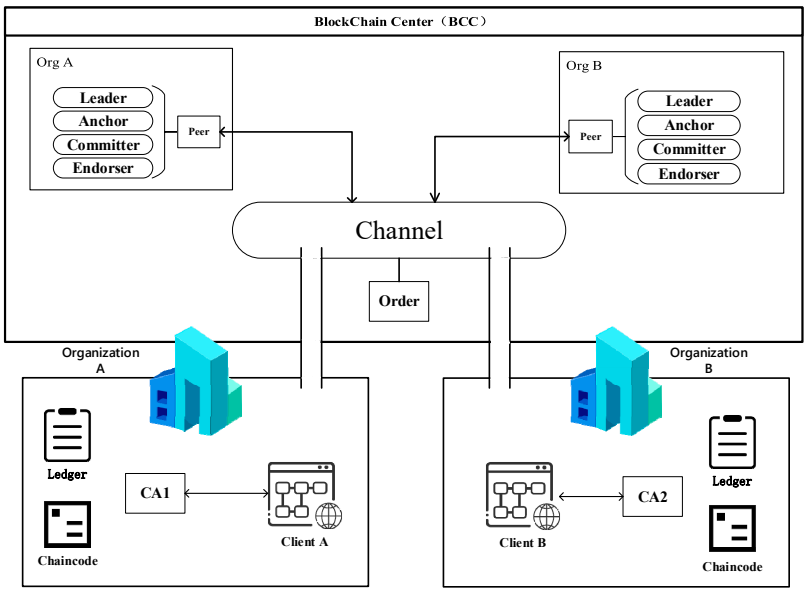

Figure 4. Trading Information Framework of Hyperledger Fabric.

Suppose that Organization A and Organization B want to interact with each other. In that case, firstly, Organization A and Organization B register with the CA node of each organization through the client of each organization, and the CA node returns the authentication certificate, public key, and private key to verify the validity of their identities. Organization A encrypts the transaction information with the distributed public key and uploads the encrypted information to the blockchain through the order node via the peer node. Organization B receives the transaction information uploaded to the blockchain by Organization A through the channel and triggers the chain code to update the local blockchain ledger for data storage. In the overall framework, the information interaction between two different functions is mainly done through channels.

3.3. Initialization Phase

During the initialization phase, each company with different roles forms an organization. Figure 5 represents the basic chain code structure in the Hyperledger Fabric network architecture that we designed. The left side of the structure stores the information of designer apparel, and the right side shows the structure and enumeration of role types of the roles participating in the supply chain. When a designer apparel product is manufactured in the supply chain, each detail is appended to the chain code structure through the roles involved in the supply chain.

```
type Clothing_Detail struct{
  C_ID string
  Clothing_information string
  Create_Datetime time.Time
  MaterialSupplier_ID string
  MaterialSupplier_Datetime DateTime
  MaterialSupplier_Factory_Name string
  Manufacturer_ID string
  Manufacturer_Datetime DateTime
  Manufacturer_Factory_Name string
  Retailer_ID string
  Retailer_Datetime DateTime
  Retailer_Sell_Datetime DateTime
  Retailer_Factory_Name string
  Retailer_ClothingPrice float
  BC_Signature string
  MS_Signature string
  MF_Signature string
  R_Signature string
}
```

```
type Roles string const{
  BrandCompany
  MaterialSupplier
  Manufacturer
  Retailer
  Customer
  ThirdParty
}

Type Roles_Information struct{
  ID string
  Name string
  Detail string
  Var RoleTypes Roles
}
```

Figure 5. Chain code of famous brand clothing structure.

3.4. Registration Phase

All nodes that want to join the Hyperledger Fabric blockchain network, including the BC, MS, MF, R, CU, and TH, need to register with the certificate authority (CA) node in the Blockchain Center, and will be given a corresponding public and private key. We use 'Roles X' to represent all arbitrary roles in the blockchain system; Figure 6 represents the flowchart of the registration phase.

Step 1. The name ID_X is generated for all roles (Roles X) participating in the anti-counterfeit and traceable management system. The generated ID_X is then sent through the application to the CA nodes in the blockchain network for registration and verification, for the legitimacy of their identities.

Step 2. The CA node in the blockchain network generates the private key k_X based on the system role and calculate the public key Q_X:

$$Q_X = d_X G \tag{1}$$

After verifying the registration of all system roles, the chain code Algorithm 1 is triggered. The CA node sends the generated (ID_X, d_X, Q_X) to Roles X.

Step 3. The role in the system receives its signature message parameter (ID_X, d_X, Q_X) and stores it.

Algorithm 1. A scheme for a chain code registration.

```
var X[]Roles X
func Registration (X_name string, X_detail string, var X_role RoleType) (C_ID string) {
      C_ID = GenerateUniqueID()
   X = append (X, Roles X{
      ID: C_ID,
      Name: X_name,
      Detail: X_detail,
      Role: X_role,
   })
      return C_ID
   }
```

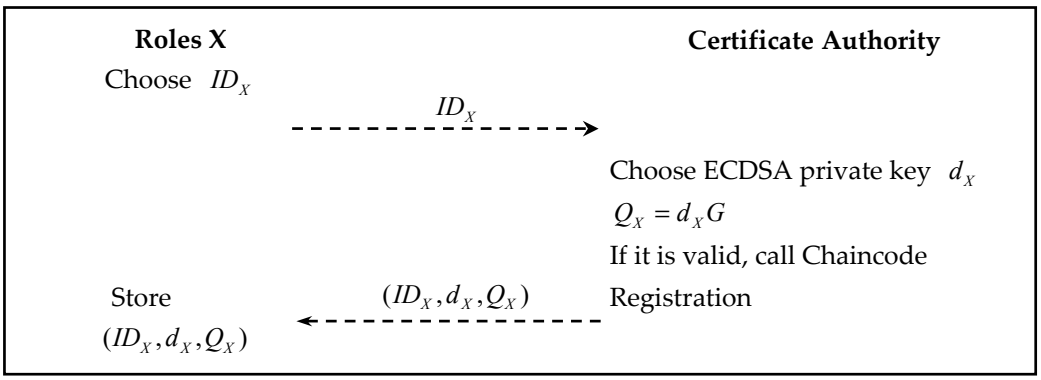

Figure 6. Registration Stage Flow Chart.

3.5. Clothing Design Phase

This phase is the clothing design phase, and the main players involved are the brand company (BC) and material supplier (MS). The flowchart is shown in Figure 7, and the related chain code is shown in Algorithms 2 and 3.

Algorithm 2. A scheme for a communication protocol.

```
func Sign (h string,k string, d string)(r string, s string){
   (x,y) = k*G;
   r = x%n
   s = (h + r*d)/x%n
   return r,s
}
func Verify(h string,r string,s string)(result string){
   u1 = h/s%n
   u2 = r/s%n
   (x,y) = u1*G + u2*Q
   If x = r{
return "valid"
   }else{
return "invalid"
   }
}
```

Algorithm 3. Chain code of the clothing design phase.

```
func BrandCompany (ID_BC string, ID_MS string, C_INF string, C_IDs []string, Signature string) {
for i:= 0; i < C_IDs.Length; i++ {
   index:= SearchC_ID(C_IDs[i]);
       TP[index].Clothing_Detail.BrandCompany_ID = ID_BC
       TP[index].Clothing_Detail..Clothing_information = C_INF
       TP[index].Clothing_Detail.Create_Datetime = time.Now()
       TP[index].Clothing_Detail.MaterialSupplier_ID = ID_MS
       TP[index].Clothing_Detail.BC_Signature = Signature
   }
}

func MaterialSupplier (ID_BC string, ID_MS string, MSname string, C_IDs []string, Signature
string) {
for i:= 0; i < C_IDs.Length; i++ {
   index:= SearchC_ID(C_IDs[i]);
       TP[index].Clothing_Detail.MaterialSupplier_ID = ID_MS
       TP[index].Clothing_Detail.MaterialSupplier_Factory_Name =MSname
       TP[index].Clothing_Detail. MaterialSupplier_Datetime = time.Now()
       TP[index].Clothing_Detail.BrandCompany_ID = ID_BC
       TP[index].Clothing_Detail.MS_Signature = Signature
   }
}
```

Step 1. When the BC wants to produce branded clothes, the MS needs to be provided with information on the raw materials required for the production of branded clothes as well as product information. The BC randomly selects a random number k_1 and generates a message containing a list of C_ID:

$$M_{BC} = (ID_{BC} \| ID_{MS} \| List < C\_ID > \| T_1) \tag{2}$$

The BC calculates its hash value and executes the "Sign" algorithm used to generate the signature (r_{BC1}, s_{BC1}); The "Sign" algorithm is shown in Algorithm 2:

$$h_{BC1} = H(M_{BC}) \tag{3}$$

$$(r_{BC1}, s_{BC1}) = Sign(h_{BC1}, k_1, d_{BC}) \tag{4}$$

Its generated encrypted messages is encrypted by the MS public key:

$$C_{BC1} = E_{Puk_{MS}}(M_{BC}) \tag{5}$$

Then BC executes the chain code function "BrandCompany", and the algorithm is shown in Algorithm 3. Send the message $(ID_{BC}, ID_{MS}, C_{BC1}, (r_{BC1}, s_{BC1}))$ to the MS.

Step 2. The MS receives the above message at a time T_2 and first decrypts the message with its private key:

$$M_{BC} = D_{prk_{MS}}(C_{BC1}) \tag{6}$$

Subsequently, MS verifies the validity of the timestamp:

$$Check(T_2 - T_1) \leq {}_{\Delta}T \tag{7}$$

The MS then uses "Verify" in Algorithm 2 to compute the hash value to verify the message:

$$h_{BC1}{}' = H(M_{BC}) \tag{8}$$

$$Sign(h_{MS1}, k_2, d_{MS}) \tag{9}$$

If the signature is valid, MS executes the chain code function "MaterialSupplier" with the algorithm shown in Algorithm 3. The MS will provide raw material and update the local ledger based on the transaction information in the encrypted message $List < C\_ID >$. The MS randomly selects a random number k_2 for generating the response message:

$$M_{MS} = (ID_{MS} \| ID_{BC} \| List < C\_ID > \| T_3) \tag{10}$$

The MS calculates its hash value and executes the "Sign" algorithm in Algorithm 2 to generate the signature (r_{MS1}, s_{MS1}):

$$h_{MS1} = H(M_{MS}) \tag{11}$$

$$(r_{MS1}, s_{MS1}) = Sign(h_{MS1}, k_2, d_{MS}) \tag{12}$$

Its generated encrypted messages are encrypted by the BC's public key:

$$C_{MS1} = E_{puk_{BC}}(M_{MS}) \tag{13}$$

Send the response message $(ID_{MS}, ID_{BC}, C_{MS1}, (r_{MS1}, s_{MS1}))$ to the BC.

Step 3. The BC receives the response message and first decrypts the message with the private key:

$$M_{MS} = D_{prk_{BC}}(C_{MS1}) \tag{14}$$

The validity of the timestamp is then confirmed by comparing:

$$Check(T_4 - T_3) \leq {}_{\Delta}T \tag{15}$$

The BC verifies the message by computing the hash value of "Verify" in Algorithm 2:

$$h_{MS1}{}' = H(M_{MS}) \tag{16}$$

$$Verify(h'_{MS1}, r_{MS1}, s_{MS1}) \tag{17}$$

If the signature is valid, the BC updates the response message from the MS to its book, and the BC executes the chain code function "BrandCompany" with the algorithm shown in Algorithm 3.

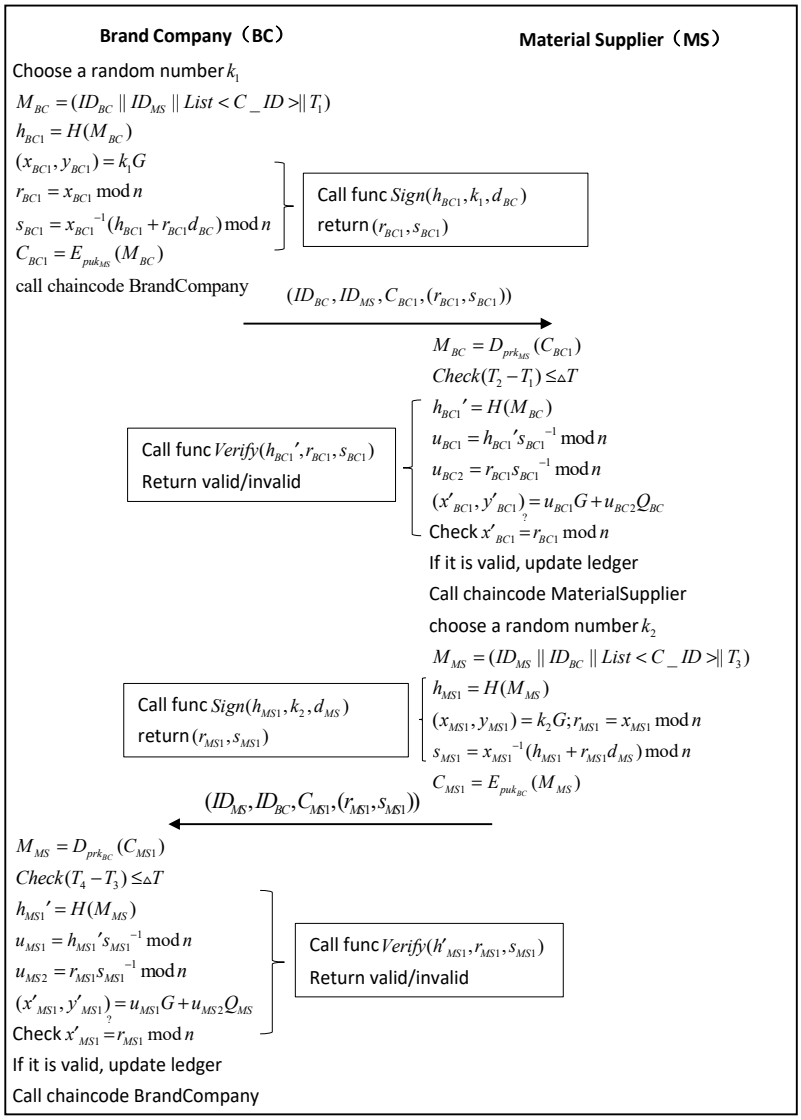

Figure 7. Clothing design phase.

After the above three steps, the clothing design phase is complete. Moreover, the raw material quantity and designer clothing design details should be updated in the account book.

3.6. Clothing Production Phase

This phase is the clothing production phase; the main players involved are the material supplier (MS) and manufacturer (MF). The flowchart is shown in Figure 8, and the related chain code is shown in Algorithm 4.

Algorithm 4. Chain code of clothing production phase.

```
func MaterialSupplier (ID_ MS string, ID_MF string, MSname string, C_IDs []string, Signature
string) {
for i:= 0; i < C_IDs.Length; i++ {
    index:= SearchC_ID(C_IDs[i]);
      TP[index].Clothing_Detail.MaterialSupplier_ID = ID_MS
       TP[index].Clothing_Detail.MaterialSupplier_Factory_Name =MSname
       TP[index].Clothing_Detail. MaterialSupplier_Datetime = time.Now()
       TP[index].Clothing_Detail. Manufacturer _ID = ID_MF
       TP[index].Clothing_Detail.MS_Signature = Signature
    }
}

func Manufacturer (ID_ MF string, ID_MS string, MFname string, C_IDs []string, Signature
string) {
for i:= 0; i < C_IDs.Length; i++ {
    index:= SearchC_ID(C_IDs[i]);
       TP[index].Clothing_Detail.Manufacturer_ID = ID_MF
       TP[index].Clothing_Detail.Manufacturer _Factory_Name =MSname
       TP[index].Clothing_Detail.Manufacturer _Datetime = time.Now()
       TP[index].Clothing_Detail.MaterialSupplier _ID = ID_MS
       TP[index].Clothing_Detail.MF_Signature = Signature
    }
}
```

Step 1. The MS provides raw materials for branded apparel to the MF for manufacturing branded clothing based on production information provided by the BC, and the MS needs to provide raw material information and production information for branded clothing to the MF. The MF produces clothes based on raw materials received in real life. The MS randomly selects a random number k_3 and generates a message containing a list of C_ID:

$$M_{MS} = (ID_{MS} \| ID_{MF} \| List < C\_ID > \| T_5) \tag{18}$$

The MS calculates its hash value and executes the "Sign" algorithm in Algorithm 2 to generate the signature (r_{MS2}, s_{MS2}):

$$h_{MS2} = H(M_{MS}) \tag{19}$$

$$(r_{MS2}, s_{MS2}) = Sign(h_{MS2}, k_3, d_{MS}) \tag{20}$$

Its generated encrypted messages are encrypted by the MF's public key:

$$C_{MS2} = E_{puk_{MF}}(M_{MS}) \tag{21}$$

Then, the MS executes the chain code function "MaterialSupplier"; the algorithm is shown in Algorithm 4. Send the production-related information of the branded clothing to MF $(ID_{MS}, ID_{MF}, C_{MS2}, (r_{MS2}, s_{MS2}))$.

Step 2. The MF receives the above message and first decrypts the message with its private key:

$$M_{MS} = D_{prk_{MF}}(C_{MS2}) \tag{22}$$

Subsequently, the MF checks the validity of the timestamp:

$$Check(T_6 - T_5) \leq {}_{\Delta}T \tag{23}$$

The MF then verifies the message by calculating the hash value with "Verify" in Algorithm 2:

$$h_{MS2}{}' = H(M_{MS}) \tag{24}$$

$$Verify(h_{MS2}{}', r_{MS2}, s_{MS2}) \tag{25}$$

If the signature is valid, the MS updates the local ledger with the transaction information in the encrypted message $List < C\_ID >$. The MF executes the chain code function "Manufacturer" with the algorithm shown in Algorithm 4. After the manufacturing of the designer clothes is completed, the MF sends a response message to the MS. The MF randomly selects a random number k_4 for generating the response message:

$$M_{MF} = (ID_{MS} \| ID_{BC} \| List < C\_ID > \| T_3) \tag{26}$$

The MF calculates its hash value and executes the "Sign" algorithm in Algorithm 2 to generate the signature (r_{MF1}, s_{MF1}):

$$h_{MF1} = H(M_{MF}) \tag{27}$$

$$(r_{MF1}, s_{MF1}) = Sign(h_{MF1}, k_4, d_{MF}) \tag{28}$$

Its generated encrypted messages are encrypted by the MS public key:

$$C_{MF1} = E_{puk_{MS}}(M_{MF})) \tag{29}$$

Send the response message $(ID_{MF}, ID_{MS}, C_{MF1}, (r_{MF1}, s_{MF1}))$ to the MS.

Step 3. The MS receives the response message and first decrypts the message with the private key:

$$M_{MF} = D_{prk_{MS}}(C_{MF1}) \tag{30}$$

The validity of the timestamp is then confirmed by calculating:

$$Check(T_8 - T_7) \leq {}_{\Delta}T \tag{31}$$

The MS verifies the message by calculating the hash value of "Verify" in Algorithm 2:

$$h_{MF1}{}' = H(M_{MF}) \tag{32}$$

$$Verify(h'_{MF1}, r_{MF1}, s_{MF1}) \tag{33}$$

If the signature is valid, the MS updates the MF's response message to its book, and the MS executes the chain code function "MaterialSupplier" with the algorithm shown in Algorithm 4.

After the above three steps, the clothing production phase is complete. The information of the produced branded clothing in the ledger should be updated.

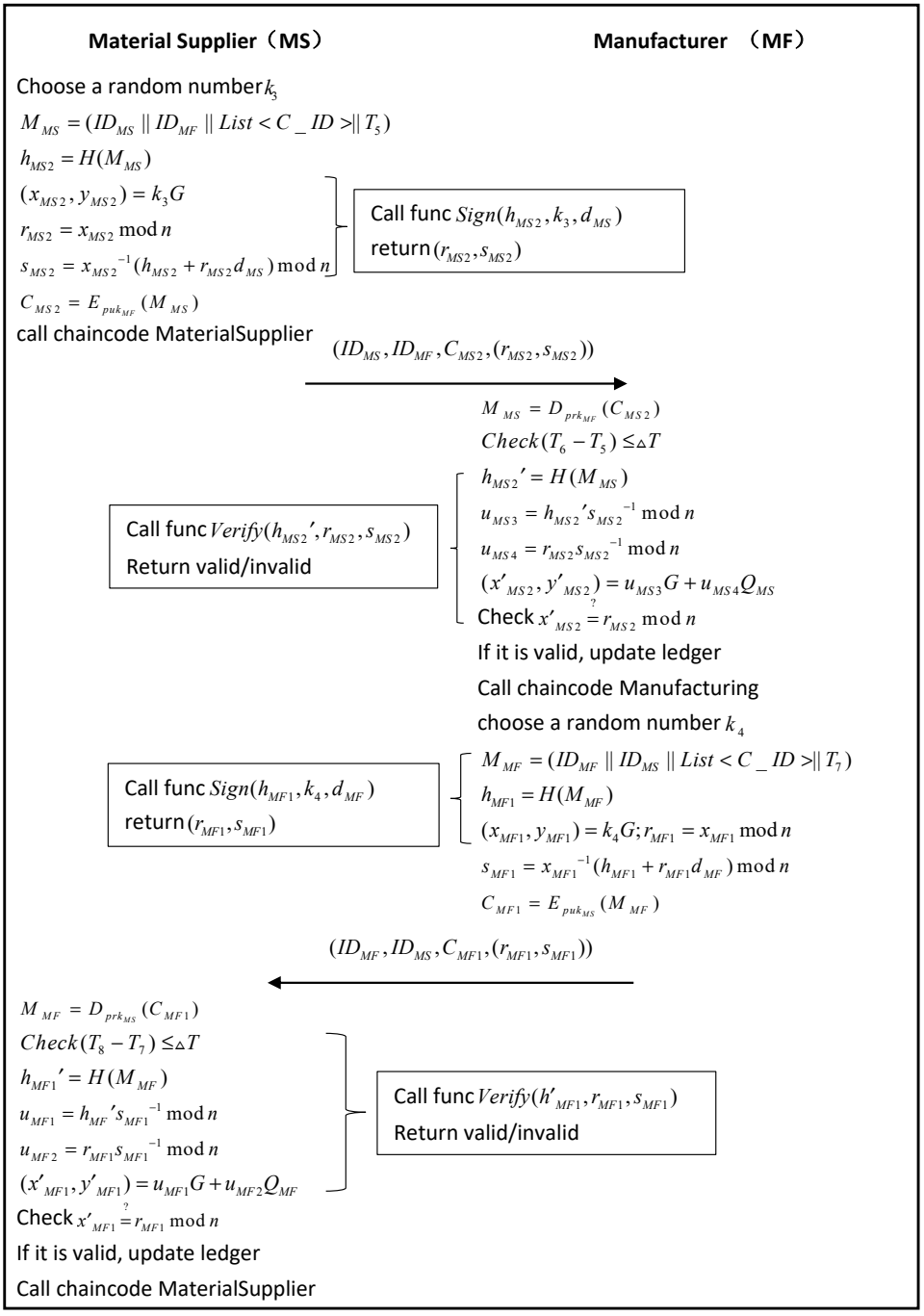

Figure 8. Clothing production phase.

3.7. Clothing Distribution Phase

This phase is the clothing production phase; the main players involved are manufacturer (MF) and the retailer (R). The flowchart is shown in Figure 9, and the related chain code is shown in Algorithm 5.

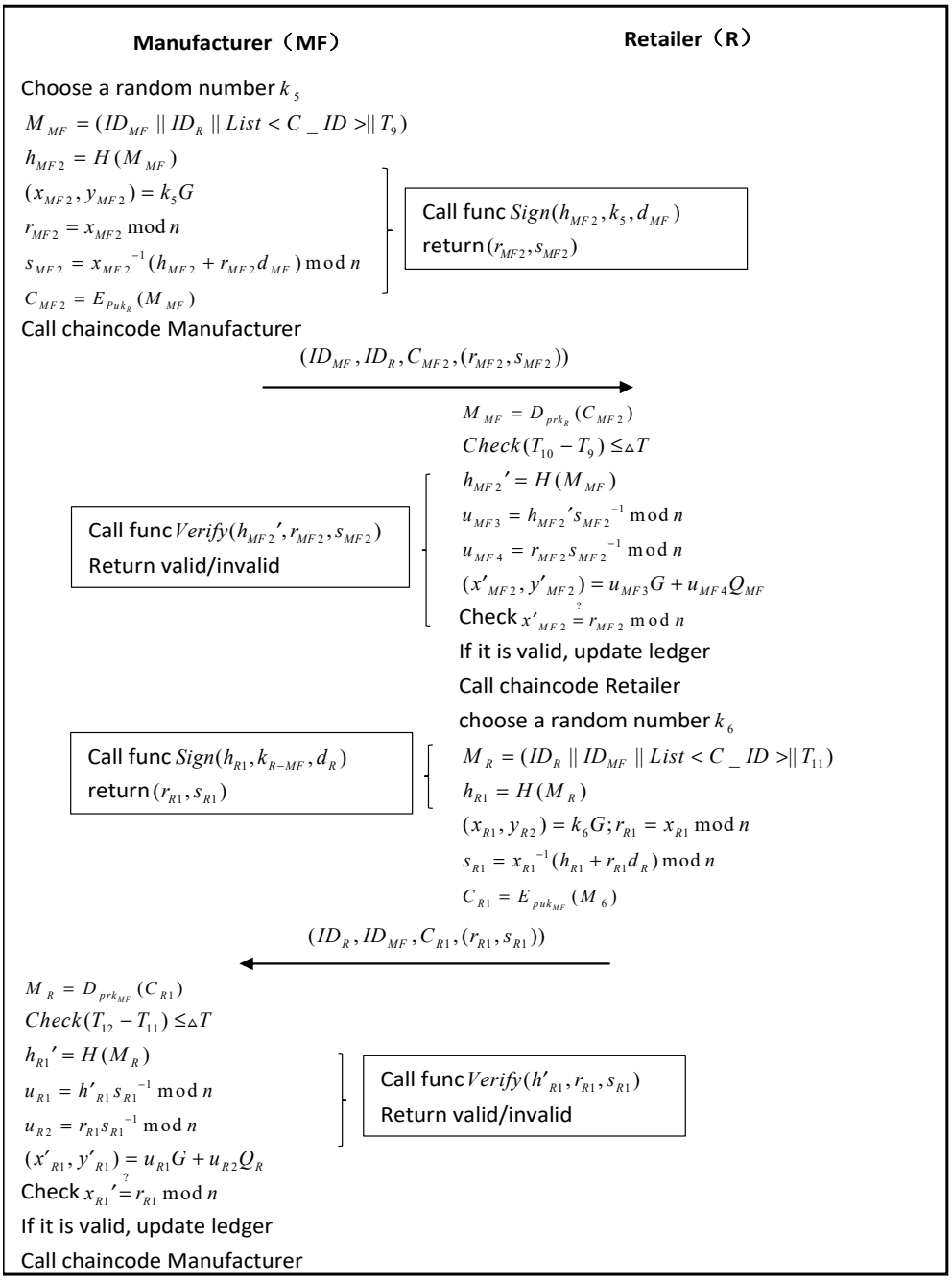

Figure 9. Clothing distribution phase.

Algorithm 5. Chain code of the clothing distribution phase.

```
func Manufacturer (ID_ MF string, ID_R string, MFname string, C_IDs []string, Signature string) {
for i:= 0; i < C_IDs.Length; i++ {
    index:= SearchC_ID(C_IDs[i]);
       TP[index].Clothing_Detail.Manufacturer_ID = ID_MF
       TP[index].Clothing_Detail.Manufacturer _Factory_Name = MSname
       TP[index].Clothing_Detail.Manufacturer _Datetime = time.Now()
       TP[index].Clothing_Detail.MaterialSupplier _ID = ID_MS
       TP[index].Clothing_Detail.MF_Signature = Signature
    }
}

func Manufacturer (ID_ R string, ID_MF string, R_sell_datetime string, C_IDs []string, Signature
string) {
for i:= 0; i < C_IDs.Length; i++ {
    index:= SearchC_ID(C_IDs[i]);
       TP[index].Clothing_Detail.Retailer_ID = ID_MF
       TP[index].Clothing_Detail.Retailer _Sell_Datetime = R_sell_datetime
       TP[index].Clothing_Detail.Retailer _Datetime = time.Now()
       TP[index].Clothing_Detail.Retailer _ID = ID_R
       TP[index].Clothing_Detail.R_Signature = Signature
    }
}
```

Step 1. The MF processes and manufactures the raw materials based on the raw material and production information provided by the MS, and sends the manufactured branded clothes to retailers for sale. The MF needs to provide branded clothing products and information on the production of branded clothing to the R. The R sells branded clothes based on the received in real life. MF randomly selects a random number k_5 and generates a message containing a list of C_ID.

$$M_{MF} = (ID_{MF} \| ID_R \| List < C\_ID > \| T_9) \tag{34}$$

The MF calculates its hash value and executes the "Sign" algorithm in Algorithm 2 to generate the signature (r_{MF2}, s_{MF2}):

$$h_{MF2} = H(M_{MF}) \tag{35}$$

$$(r_{MF2}, s_{MF2}) = Sign(h_{MF2}, k_5, d_{MF}) \tag{36}$$

Its generated encrypted messages are encrypted by the R's public key:

$$C_{MF2} = E_{Puk_R}(M_{MF}) \tag{37}$$

Then, the MF executes the chain code function "Manufacturer"; the algorithm is shown in Algorithm 5. Send the information $(ID_{MF}, ID_R, C_{MF2}, (r_{MF2}, s_{MF2}))$ related to the production of brander clothes to the R.

Step 2. The R receives the above message and first decrypts the message with its private key:

$$M_{MF} = D_{prk_R}(C_{MF2}) \tag{38}$$

The R will then check the validity of the timestamp:

$$Check(T_{10} - T_9) \leq {}_{\Delta}T \tag{39}$$

Then, the R verifies the message by calculating the hash value with "Verify" in Algorithm 2:

$$h_{MF2}{}' = H(M_{MF}) \tag{40}$$

$$Verify(h_{MF2}{}', r_{MF2}, s_{MF2}) \tag{41}$$

If the signature is valid, the R updates the local ledger based on the transaction information in the encrypted message $List < C\_ID >$. The R executes the chain code function "Retailer" with the algorithm shown in Algorithm 5. After the retailer receives the branded clothes products from the manufacturer, the R sends a response message to the MF. The R randomly selects a random number k_6 for generating the response message:

$$M_R = (ID_R \| ID_{MF} \| List < C\_ID > \| T_{11}) \tag{42}$$

The R calculates its hash value and executes the "Sign" algorithm in Algorithm 2 to generate the signature (r_{R1}, s_{R1}):

$$h_{R1} = H(M_R) \tag{43}$$

$$(r_{R1}, s_{R1}) = Sign(h_{R1}, k_6, d_R) \tag{44}$$

Its generated encrypted messages is encrypted by the MF's public key:

$$C_{R1} = E_{puk_{MF}}(M_R) \tag{45}$$

Send the response message $(ID_R, ID_{MF}, C_{R1}, (r_{R1}, s_{R1}))$ to the MF.

Step 3. The MF receives the response message and first decrypts the message with the private key:

$$M_R = D_{prk_{MF}}(C_{R1}) \tag{46}$$

Then to confirm the validity of the timestamp:

$$Check(T_{12} - T_{11}) \leq {}_{\Delta}T \tag{47}$$

The MF verifies the message by calculating the hash value of "Verify" in Algorithm 2:

$$h_{R1}{}' = H(M_R) \tag{48}$$

$$Verify(h'_{R1}, r_{R1}, s_{R1}) \tag{49}$$

If the signature is valid, the MF updates the response message from the R to its book, and the MF performs the chain code function "Manufacturer" to the chain and updates the book information; the algorithm is shown in Algorithm 5.

After the above three steps, the clothing distribution phase is complete. The information of the designer apparel received by the retailer in the ledger should be updated.

3.8. User Authentication Phase

After a consumer receives the branded clothing item, to verify the legitimacy of his/her branded clothes, the consumer can connect to the blockchain hub via an application to query the information about the apparel, to confirm the correctness of the apparel. The process is shown in Figure 10.

Step 1. Customers use the application to search for consumption information, such as the number of the designer apparel (C_ID), the designer apparel company logo, the manufacturer logo, and the purchase time stamp.

Step 2. The client can call the GetHistoryForKey () method of the ChaincodeStubInterface interface to query its history. Its chain code algorithm is shown in Algorithm 6 to query the ledger.

Algorithm 6. Call the chaincode to query the ledger

```
historyIter, err := stub.GetHistoryForKey(yourKey)
if err != nil {
fmt.Println(errMsg)
return shim.Error(errMsg)
}
if historyIter.HasNext() {
modification, err := historyIter.Next()
if err != nil {
fmt.Println(errMsg)
return shim.Error(errMsg)
}
fmt.Println("Returning information related to", string(modification.Value))
```

Step 3. If the C_ID provided by the client is legitimate and valid, then the blockchain center will return the ledger information associated with it. If the C_ID provided cannot be queried in the blockchain center's ledger, we could confirm it is an illegally counterfeited brand-name clothing item.

Step 4. The APP program displays the results of the chain code, in which consumers can get the legitimacy and production information of the branded clothing products to confirm whether they are buying the legal and correct goods.

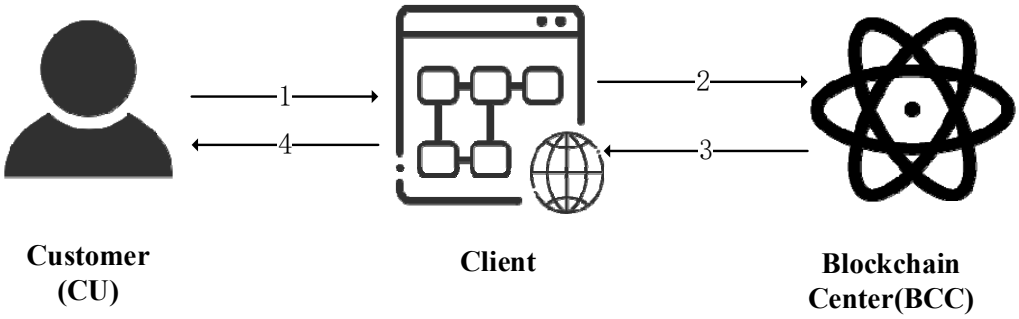

Figure 10. Flow chart of user query information.

3.9. Arbitration Phase

In fact, when any of the actors involved in the supply chain have doubts about the authenticity of the designer apparel, they can arbitrate the system's legitimacy through a third party arbiter. Among the arbiters are the four supply chain sectors of the federated chain and the consumer. The third party validates the arbitration requests of its actors; the third party validation phase is shown in Figure 11.

Step 1. All participants provide information, such as the number (C_ID) and signature message of the designer apparel to the third party to query and verify the legitimacy of the designer apparel.

Step 2. The third party sends a request with the participant's signature information and C_ID to the blockchain center via its C_ID.

Step 3. The blockchain center verifies the legitimacy of its signature and returns a list of its corresponding messages if it is legitimate.

Step 4. The third party checks the signature and the steps are:

- The third party will first collect signatures and data.
- Check the signature of the retailer (R), and if the R's signature is illegal, determine that the R falsified the record.
- If the signature of the retailer (R) is legal, the signature of the manufacturer (MF) is examined, and if the MF's signature is illegal, the MF is judged to have falsified the record.

- After the signature of the manufacturer (MF) is deemed legal, the signature of the material supplier (MS) is checked, and if the signature of the MS is illegal, the MS is judged to have falsified the record.
- If the signature of the material supplier (MS) is legal, then inspect the signature of the brand company (BC), and if the BC's signature is illegal, determine that the BC falsified the record. If all signatures are valid and legal, the third party determines that there are no illegal records.

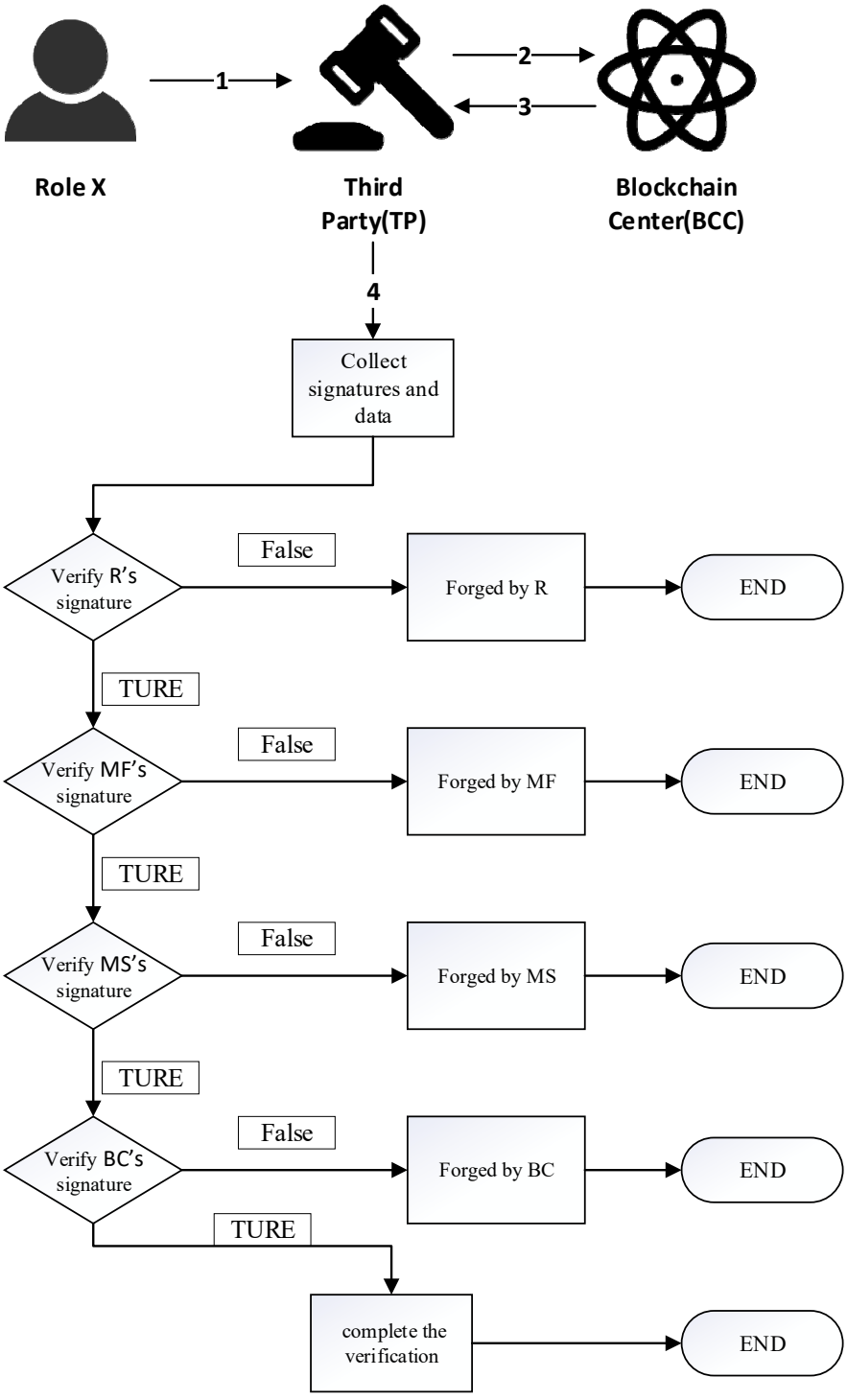

Figure 11. Arbitration phase flow chart.

4. Security Analysis

4.1. Mutual Authentication

In the mutual authentication phase, we use BAN logic for mutual authentication between two participants, where we first list the relevant symbols of BAN logic, as follows:

$P\|\equiv X$	P believes X (belief rule)
$P \lhd X$	P sees X (seeing rule)
$P\|\sim X$	P once said X (message meaning rule)
$P\|\Rightarrow X$	P has jurisdiction over X (jurisdiction rule)
$\#(X)$	The message X is new (freshness rule)
$\{X\}_K$	The message X is encrypted by a key K
$P \overset{x}{\leftrightarrow} Q$	P and Q use a shared key x to communicate

We take user A and user B as an example in this phase. We mainly use BAN logic to make the authentication between company A and company B. The objectives of the authentication analysis are shown below.

$$G1: A\left|\equiv A \overset{K_{B-A}}{\leftrightarrow} B\right.$$

$$G2: A\left|\equiv B\right|\equiv A \overset{K_{B-A}}{\leftrightarrow} B$$

$$G3: B\left|\equiv A \overset{K_{A-B}}{\leftrightarrow} B\right.$$

$$G4: B\left|\equiv A\right|\equiv A \overset{K_{A-B}}{\leftrightarrow} B$$

$$G5: A|\equiv ID_B$$

$$G6: A|\equiv B|\equiv ID_B$$

$$G7: B|\equiv ID_A$$

$$G8: B|\equiv A|\equiv ID_A$$

According to the BAN logic authentication algorithm, the following ideal form is generated:

$$M1: A \rightarrow B(\{ID_A, ID_B, T_{A-B}, M_{A-B}\}_{puk_B}, < h(ID_A, ID_B, T_{A-B}, M_{A-B}) >_{x_{A-B}})$$

$$M2: B \rightarrow A(\{ID_B, ID_A, T_{B-A}, M_{B-A}\}_{puk_A}, < h(ID_B, ID_A, T_{B-A}, M_{B-A}) >_{x_{A-B}})$$

We present the following assumptions to analyze the proposed program:

$$A1: A|\equiv \#(k_{X-Y})$$

$$A2: B|\equiv \#(k_{X-Y})$$

$$A3: A|\equiv \#(k_{Y-X})$$

$$A4: B|\equiv \#(k_{Y-X})$$

$$A5: A\left|\equiv B\right|\Rightarrow A \overset{x_{Y-X}}{\leftrightarrow} B$$

$$A6: B\left|\equiv A\right|\Rightarrow B \overset{x_{X-Y}}{\leftrightarrow} A$$

$$A7: A|\equiv B|\Rightarrow ID_B$$

$$A8: B|\equiv A|\Rightarrow ID_A$$

a. User B authenticates User A By M1 and the seeing rule, we can derive:
$B \lhd (\{ID_A, ID_B, T_{A-B}, M_{A-B}\}_{puk_B}, < h(ID_A, ID_B, T_{A-B}, M_{A-B}) >_{x_{A-B}})$ (Statement 1)
By A2 and the freshness rule, we can derive:
$B\left|\equiv \#(\{ID_A, ID_B, T_{A-B}, M_{A-B}\}_{puk_B}, < h(ID_A, ID_B, T_{A-B}, M_{A-B}) >_{x_{A-B}})\right.$
(Statement 2)
By (Statement 1) and the message meaning rule, we can derive:
$B|\equiv A|\sim (\{ID_A, ID_B, T_{A-B}, M_{A-B}\}_{puk_B}, < h(ID_A, ID_B, T_{A-B}, M_{A-B}) >_{x_{A-B}})$ (Statement 3)
By (Statement 2), (Statement 3), and the nonce verification rule, we can derive:
$B\left|\equiv A\right|\equiv (\{ID_A, ID_B, T_{A-B}, M_{A-B}\}_{puk_B}, < h(ID_A, ID_B, T_{A-B}, M_{A-B}) >_{x_{A-B}})$ (Sta-

tement 4)
By (Statement 4) and the belief rule, we can derive (G4):
$B\left|\equiv A\right|\equiv A \overset{x_{A-B}}{\leftrightarrow} B$ (Statement 5)
By (Statement 5), A6, and the jurisdiction rule, we can derive (G3):
$B\left|\equiv A \overset{x_{A-B}}{\leftrightarrow} B\right.$ (Statement 6)
By (Statement 4) and the belief rule, we can derive (G8):
$B|\equiv A|\equiv ID_A$ (Statement 7)
By (Statement 7), A8, and the belief rule, we can derive (G7):
$B|\equiv ID_A$ (Statement 8)

b. User B authenticates User A By M2 and the seeing rule, we can derive:
$A \lhd (\{ID_B, ID_A, T_{B-A}, M_{B-A}\}_{puk_A}, < h(ID_B, ID_A, T_{B-A}, M_{B-A}) >_{x_{B-A}})$ (Statement 9)
By A3 and the freshness rule, we can derive:
$A\left|\equiv \#(\{ID_B, ID_A, T_{B-A}, M_{B-A}\}_{puk_A}, < h(ID_B, ID_A, T_{B-A}, M_{B-A}) >_{x_{B-A}})\right.$ (Statement 10)
By (Statement 9) and the message meaning rule, we can derive:
$A|\equiv B| \sim (\{ID_B, ID_A, T_{B-A}, M_{B-A}\}_{puk_A}, < h(ID_B, ID_A, T_{B-A}, M_{B-A}) >_{x_{B-A}})$ (Statement 11)
By (Statement 10), (Statement 11) and the nonce verification rule, we can derive:
$A\left|\equiv B\right|\equiv (\{ID_B, ID_A, T_{B-A}, M_{B-A}\}_{puk_A}, < h(ID_B, ID_A, T_{B-A}, M_{B-A}) >_{x_{B-A}})$ (Statement 12)
By (Statement 12) and the belief rule, we can derive (G2):
$A|\equiv B|\equiv ID_B$ (Statement 13)
By (Statement 13), A5, and the jurisdiction rule, we can derive (G1):
$A\left|\equiv A \overset{x_{B-A}}{\leftrightarrow} B\right.$ (Statement 14)
By (Statement 12) and the belief rule, we can derive (G6):
$A|\equiv B|\equiv ID_B$ (Statement 15)
By (Statement 15), A7, and the belief rule, we can derive (G5):
$A|\equiv ID_B$ (Statement 16)

According to (Statement 6), (Statement 8), (Statement 14), and (Statement 16), we can prove that mutual authentication between user A and user B is possible. Two different users can authenticate their identities to each other.

4.2. Data Integrity

In this study, we used the Elliptic Curve Encryption Algorithm (ECDSA) to sign the message transmission between participants, to ensure the integrity and 'tamper proofness' of the message transmission process. We use the clothing design phase as an example; when the designer apparel company wants to send a message to the raw material provider for information interaction, the designer apparel company generates the ECDSA signed value (r_{BC1}, s_{BC1}). The branded clothing company sends its public key Q_{BC}, digital signature (r_{BC1}, s_{BC1}), and a message C_{BC1} encrypted with the manufacturer's public key to the manufacturer. An external attacker cannot obtain the manufacturer's private key because it is not available. Thus, it is impossible to attack the data after it has undergone ECDSA signing. The manufacturer receives the message C_{BC1} and decrypts it with its private key. Calculation:

$h_{BC1}{}' = H(M_{BC})$
$u_{BC1} = h_{BC1}{}' {s_{BC1}}^{-1} \mathrm{mod} n$
$u_{BC2} = r_{BC1} {s_{BC1}}^{-1} \mathrm{mod} n$
$(x'_{BC1}, y'_{BC1}) = u_{BC1} G + u_{BC2} Q_{BC}$

Determine if the data are complete by comparing whether x'_{BC1} and $r_{BC1} \mathrm{mod} n$ are equal.

From the above description, we can see that this study uses the ECDSA signature technique to ensure the integrity of the data during transmission effectively.

4.3. Non-Repudiation

Since each stage requires signature verification by ECDSA, we can achieve the problem of data non-repudiation by verifying ECDSA. Since each data transmission requires the signature of its private key, the receiver also needs the public key for verification. The receiver will not reject the content of the message sent by the sender after verifying the correctness and legitimacy of the message. Table 2 shows the non-repudiation in each stage.

Table 2. Non-repudiation of the proposed scheme.

Phase \ Item	Signature Value	Sender	Receiver	Signature Verification
Clothing design phase	(r_{BC1}, s_{BC1})	BC	MS	$x'_{BC1} \stackrel{?}{=} r_{BC1} \bmod n$
	(r_{MS1}, s_{MS1})	MS	BC	$x'_{MS1} \stackrel{?}{=} r_{MS1} \bmod n$
Clothing production phase	(r_{MS2}, s_{MS2})	MS	MF	$x'_{MS2} \stackrel{?}{=} r_{MS2} \bmod n$
	(r_{MF1}, s_{MF1})	MF	MS	$x'_{MF1} \stackrel{?}{=} r_{MF1} \bmod n$
Clothing distribution phase	(r_{MF2}, s_{MF2})	MF	R	$x'_{MF2} \stackrel{?}{=} r_{MF2} \bmod n$
	(r_{R1}, s_{R1})	R	MF	$x'_{R1} \stackrel{?}{=} r_{R1} \bmod n$

4.4. Resist Known Attacks

4.4.1. Man-in-the-Middle Attack

The man-in-the-middle attack means that, after the sender of a transaction comes through the blockchain to send a transaction, the receiver has not confirmed it as yet. By modifying the data in the transaction, the attacker can change the data in the transaction into a new transaction. This can make the transaction received by the receiver different from the one sent by the sender. In severe cases, the transaction will be considered illegitimate and will not be validated. We prevent man-in-the-middle attacks by adding encryption and decryption mechanisms to the communication protocol. These encryptions and decryptions are shown in the following equations. Equations (5), (6), (13), (14), (21), (22), (29), (30), (37), (38), (45) and (46). For example, in the clothing design phase, when the designer apparel company wants to send a message to the raw material provider for information interaction, the BC encrypts it using the public key of the MS, and the MS decrypts it after receiving the encrypted message with the private key of the MS. The related equation is:

$$C_{BC1} = E_{puk_{MS}}(M_{BC})$$

$$M_{BC} = D_{prk_{MS}}(C_{BC1})$$

Scenario: the attacker eavesdrops or tampers with the message sent by the sender and then changes the content of the message before sending it to the receiver.

Analysis: the sender encrypts the message with the public key of the receiver, and the receiver can only decrypt the message if it has its associated private key. The attacker cannot decrypt the message because it does not have the receiver's private key and, thus, cannot tamper with the message to perform a man-in-the-middle attack.

4.4.2. Replay Attack

To prevent replay attacks, we add information, such as timestamps, when verifying messages. The timestamp needs to be verified in each data transmission phase, and the verification process is shown in the following equations. Equations (7), (15), (23), (31), (39) and (47). For example, in the clothing design phase, when the designer apparel company wants to send a message to the raw material provider for information interaction, the

designer apparel company will add a timestamp to the message when sending the message, where the timestamp is unique. BC adds the timestamp to the message M. Then, the message is encrypted and MS de-crypts it after receiving the encrypted message. The validity of the timestamp is then checked. The relevant equation is:

$$M_{BC} = (ID_{BC}||ID_{MS}||List < C\_ID >||T_1)$$

$$C_{BC1} = E_{puk_{MS}}(M_{BC})$$
$$M_{BC} = D_{prk_{MS}}(C_{BC1})$$
$$Check(T_2 - T_1) \leq {}_{\Delta}T$$

Scenario: after the attacker eavesdrops on the message sent by the sender, the attacker sends the same message to the receiver to perform a replay attack.

Analysis: the receiver decrypts the received encrypted message to get the unique timestamp in the message and then subtracts the timestamp from the current time, and if the timestamp is invalid, it is determined to be a replay attack.

4.5. Privacy Protection

For data privacy protection at the ledger level, Hyperledger Fabric invokes the concept of a channel. The role of a channel isolates the scope of data flow in the blockchain network, and members of organizations not within the channel have no access to data within the channel. Within the same Hyperledger Fabric federation network, participants can create multiple different channels according to their business needs, thus ensuring that data flows only within a specific organization. The introduction of channels greatly protects the privacy of data. The participants authenticate and register by sending them ID_X to the CA node. If the CA verifies the legitimacy of the participant, it will return its corresponding public-private key (d_X, Q_X).

Scenario: Non-supply chain participants want to access information in the supply chain for illegal access to information.

Analysis: Non-supply chain participants cannot join the channel by authenticating with CA nodes, as they cannot join the channel. Therefore, the data in the channel cannot be accessed. Thus, the privacy of information is protected.

4.6. Traceability and Tamper-Proof

After the information of famous brand clothing is uploaded to the blockchain, the information about famous brand clothing in the ledger will be kept in the blockchain forever and cannot be tampered with. All the production process and transaction information related to brand clothing will be tracked throughout the process. For example, in the clothing design phase, when the brand company wants to send a message to the raw material provider for information interaction, we can compare and verify if the blockchain data between BC and MS is legitimate if we want to verify and track it. The equation is:

$$(r_{BC1}, s_{BC1}) = Sign(h_{BC1}, k_1, d_{BC})$$

$$(r_{MS1}, s_{MS1}) = Sign(h_{MS1}, k_2, d_{MS})$$

Scenario: the customer found a manufacturing flaw in the brand clothing, but could not identify where the error had occurred.

Analysis: use the traceability and immutability in blockchain to verify and compare information in blockchain to ensure that it is traceable.

5. Discussion

5.1. Communication Analysis

In Table 3, we analyze the communication efficiency of the system. Communication efficiency analysis includes the registration phase, clothing design phase, clothing production

phase, and clothing distribution phase. Due to the different communication environments, we will discuss 3G, 4G, and 5G separately. The maximum transmission speed of 3G is 6 Mbps, 4G is 100 Mbps, and the maximum transmission speed of 5G is 20 Gbps.

Table 3. Communication cost comparison table.

	Message Length	**3G (6 Mbps)**	**4G (100 Mbps)**	**5G (20 Gbps)**
Registration phase	800 bits	133 μs	8 μs	0.040 μs
Clothing design phase	4960 bits	827 μs	50 μs	0.248 μs
Clothing production phase	4640 bits	773 μs	50 μs	0.232 μs
Clothing distribution phase	5300 bits	883 μs	53 μs	0.265 μs

By analyzing the communication protocol, we assume that the identity information (ID) requires 144 bits, the cryptographic message requires at least 512 bits, and the signed message requires 1024 bits. Let us take the clothing distribution phase as an example. The MF sends two IDs, a signed message and an encrypted message. The encrypted message includes two IDs and timestamps and two others. The total size is 2 × 144 bits + 1 × 1024 bits + 2 × 144 bits + 1 × 80 bits + 2 × 320 bits = 2320 bits. The response message sent by the MS consists of two IDs, a signature message, and an encrypted message. The encrypted message includes two IDs, a timestamp, and four others. Total size is 2 × 144 bits + 1 × 1024 bits + 2 × 144 bits + 1 × 80 bits + 4 × 320 bits = 2980 bits. The clothing distribution phase total is 2320 bits + 2980 bits = 5300 bits. We can conclude that the transmission of all messages in 3G requires 883 μs; in 4G, it costs 53 μs; while in 5G, it only costs 0.265 μs.

5.2. Computation Cost

In this study, we analyze the calculated costs. In each phase, each participant needs to perform asymmetric encryption and decryption using ECDSA. This includes a series of calculations, such as calculating hash functions, addition, subtraction, multiplication, and division as the basis for calculating cost calculations. We compare the computational costs of each phase in Table 4 for the effective perception of computational costs.

Table 4. Calculated cost comparison.

Phase	**Role A**	**Role B**
Clothing production phase	Brand company: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$	Material supplier: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$
Clothing production phase	Material supplier: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$	Manufacturer: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$
Clothing distribution Phase	Manufacturer: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$	Retailer: $2T_{asy} + 2T_h + 2T_{add} + 1T_{sub} + 4T_{mul} + 3T_{div}$

Notes: T_{asy}: asymmetrical encryption/decryption; T_h: a hash operation; T_{add}: an additional operation; T_{sub}: a subtraction operation; T_{mul}: a multiplication operation; T_{div}: a division operation.

5.3. Blockchain Architecture Comparison

There are many blockchain architectures on the market (it is an evolving technology). From the earliest bitcoin, Ethereum, to the present Hyperledger Fabric. Each blockchain architecture has its own characteristics. Table 5 presents a comparison of the main architectures in blockchain.

Table 5. Comparison table of main features of blockchain architecture.

	Bitcoin	**Ethereum**	**Hyperledger Fabric**
Consensus algorithm	POW	POW	Mainly for PBFT (practical byzantine fault tolerance)
Scene	Public chain	Public chain or Federation chain	Federation chain
Development language	C++	GO	GO
Smart contract	No	Yes	Yes
Transaction per second	7	25	100 K

Through the comparison table above, we can easily see that Hyperledger Fabric is more suitable for the application of the federation chain. Hyperledger Fabric has high TPS and supports smart contract deployment. Hyperledger Fabric is the most appropriate blockchain architecture for our proposed framework.

5.4. Comparison

In Table 6, we compare our previous solution with the one proposed in this paper. The focus of our proposed solution is to create traceable anti-counterfeit management of branded clothing. The non-repudiation, privacy, and integrity of the information are guaranteed in all aspects. For security during the process of information transmission, we also make a guarantee of information security by referring to cryptography. We further improved the anti-counterfeit traceability technology by learning from previous solutions.

Table 6. Comparison of the tobacco products logistics system.

Authors	**Year**	**Objective**	**1**	**2**	**3**	**4**	**5**
N. Alzahrani et al. [6]	2018	Block-supply chain: a new anti-counterfeiting supply chain Using NFC and blockchain	Y	Y	Y	Y	N
P. Zhu et al. [7]	2020	A blockchain-based solution for medication Anti-counterfeiting and traceability	Y	Y	N	Y	Y
J Bullón Pérez et al. [8]	2010	Traceability of ready-to-wear clothing through Blockchain technology	Y	Y	Y	N	Y
Neo C.K. et al. [9]	2021	Toward blockchain-enabled supply chain Anti-counterfeiting and traceability	Y	Y	Y	Y	N
T.K. Agrawal et al. [10]	2021	Blockchain-based framework for supply chain traceability: a case example Of textile and clothing industry	Y	Y	Y	N	N
Our Scheme	2021	An anti-counterfeit and traceable management system for brand clothing with the Hyperledger Fabric framework	Y	Y	Y	Y	Y

Notes: (1) a system framework is proposed; (2) traceability; (3) privacy protection; (4) security analysis; (5) communication security; (Y) Yes; (N) No.

We made some comparisons with previous studies, absorbed the advantages, and made improvements. We put forward a complete system framework and continued the characteristics of blockchain technology. Compared with previous studies, we added security analysis to the framework to discuss whether data are secure or not. It is not hard to see how well we have implemented data privacy and traceability, and used digital signature technology to secure communications in the supply chain. Combined with internet of things radio frequency technology, customers can easily query the production information of brand clothing for anti-counterfeiting traceability.

6. Conclusions

We proposed a Hyperledger Fabric-based anti-counterfeit management system for traceable designer apparel. We contributed to the anti-counterfeiting of brand-name clothing. In this paper, we proposed a system framework that combined blockchain and supply chain, and analyzed and explained each process in the supply chain in detail. The production and sales processes are recorded in the blockchain network so that customers could easily trace the anti-counterfeiting of the purchased branded clothing through the blockchain network. The third party arbitration mechanism can also be used to check the illegal part of the supply chain easily. It provides adequate protection for the whole production parties and customers of brand clothing.

We used the Elliptic Curve Encryption Algorithm (ECDSA) to ensure the security of the entire system. We encrypted the communication in every process in the supply chain by ECDSA to ensure the system's data integrity, non-repudiation, and privacy. By deploying and designing the chain code, we also further secured the data uploading operation and updating in the system. In particular, we used BAN logic to analyze the authentication of the identity.

By comparing with the contributions of previous solutions, this system focuses more on the security protection of data. We elaborated the system framework and analyzed the system security in all aspects through cryptography. The same analysis for communication also indicates that the system has better performance in the communication process.

In summary, the contributions made in this paper are as follows:

1. We propose a Hyperledger Fabric-based anti-counterfeit management system for traceable branded clothing, which provides a comprehensive plan for the design, production, distribution, and sale of branded clothing. It ensures the security of the data in the supply chain.
2. With the chain code algorithm to maintain and constrain the ledger, designer apparel products are difficult to forge by malicious attackers.
3. The identity verification stage is designed in the supply chain to stop counterfeit substitution by illegal companies for the production process.
4. A unique identification code is added for customers to check and analyze the authenticity of clothing, and for facilitates to query and update the data in the blockchain.
5. We use BAN logic to prove the security of inter-company authentication.

Author Contributions: Conceptualization, C.-L.C. and X.S.; methodology, C.-L.C., X.S. and Y.-Y.D.; validation, W.W., Y.-Y.D., C.-M.W. and J.C.; investigation, C.-L.C., X.S. and Y.-Y.D.; data analysis, X.S., W.W., Y.-Y.D., C.-M.W. and J.C.; writing—original draft preparation, C.-L.C. and X.S.; writing—review and editing, W.-J.T., W.W., Y.-Y.D., C.-M.W. and J.C.; supervision, C.-L.C. and W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under contract MOST 110-2218-E-305-001–MBK and contract MOST 110-2410-H-324-004-MY2, the Education and Teaching Reform Project of the Xiamen University of Technology (no. JG2021007), and the National Natural Science Foundation of China (no. 51808474).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Notations

ID_X	The Identity of X
C_ID_i	C_ID_i is the ith identity of brand clothes for each brand clothing
$List < C\_ID >$	$List < C\_ID >$ is a set of $C\_ID_1, C\_ID_2, \ldots, C\_ID_N$.
E	The elliptic curve defined on finite group
G	A generating point based on the elliptic curve E
k_i	The ith random value on the elliptic curve
d_X	The ECDSA's private key of the party X
Q_X	The ECDSA's public key of the party X
(r_{Xi}, s_{Xi})	Elliptic curve signature value of X
(x_{Xi}, y_{Xi})	An ECDSA signature value of X
$E_{puk_X}(M)/E_{prk_X}(M)$	Encrypt/decrypt the message M with a public key or private key of the party X
$H(M)$	The hash value of a message M is calculated by a one-way hash function
h_{Xi}	The ith hash value of X
T_i	The ith timestamp
$_{\Delta}T$	The threshold for checking the validity of a timestamp
M_{BC}	The message (clothing information and create datetime) from the brand company
M_{MS}	The message (material supplier datetime and material supplier name) from the material supplier
M_{MF}	The message (manufacturer datetime and manufacturer name) from the manufacturer
M_R	The message (retailer datetime, retailer sell datetime, retailer name, and clothing price) from the retailer
$A1 \stackrel{?}{=} A2$	Verify if $A1$ is equal to $A2$ or not

References

1. Modgil, S.; Sonwaney, V. Planning the application of blockchain technology in identification of counterfeit products: Sectorial prioritization. *IFAC-Pap. Line* **2019**, *52*, 1–5. [CrossRef]
2. Counterfeit Luxury Goods | World Trademark Review. Available online: https://www.worldtrademarkreview.com/anti-counterfeiting/counterfeit-luxury-goods (accessed on 20 September 2021).
3. What Impact Do Counterfeits Have on the Fashion Industry? Available online: https://www.redpoints.com/blog/fashion-counterfeit-impact/ (accessed on 20 September 2021).
4. Li, C.T.; Shih, D.H.; Wang, C.C.; Chen, C.L.; Lee, C.C. A blockchain based data aggregation and group authentication scheme for electronic medical system. *IEEE Access* **2020**, *8*, 173904–173917. [CrossRef]
5. Ding, Q.; Gao, S.; Zhu, J. Permissioned blockchain-based double-layer framework for product traceability system. *IEEE Access* **2019**, *8*, 6209–6225. [CrossRef]
6. Rui-lin, Z. Research on anti-counterfeiting technology for clothing. *Silk* **2011**, *11*, 26–28.
7. Alzahrani, N.; Bulusu, N. Block-supply chain: A new anti-counterfeiting supply chain using NFC and blockchain. In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Munich, Germany, 15 June 2018; pp. 30–35. [CrossRef]
8. Zhu, P.; Hu, J.; Zhang, Y. A blockchain based solution for medication anti-counterfeiting and traceability. *IEEE Access* **2020**, *8*, 184256–184272. [CrossRef]
9. Bullón, P.J.; Queiruga-Dios, A.; Gayoso Martínez, V. Traceability of ready-to-wear clothing through blockchain technology. *Sustainability* **2020**, *12*, 7491. [CrossRef]
10. Yiu, N.C.K. Toward Blockchain-Enabled Supply Chain Anti-Counterfeiting and Traceability. *Future Internet* **2021**, *13*, 86. [CrossRef]
11. Agrawal, T.K.; Kumar, V.; Pal, R. Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry. *Comput. Ind. Eng.* **2021**, *154*, 107130. [CrossRef]
12. Crosby, M.; Pattanayak, P.; Verma, S. Blockchain technology: Beyond bitcoin. *Appl. Innov.* **2016**, *2*, 71.
13. Chen, C.L.; Deng, Y.Y.; Li, C.T.; Zhu, S.; Chiu, Y.J.; Chen, P.Z. An IoT-based traceable drug anti-counterfeiting management system. *IEEE Access* **2020**, *8*, 224532–224548. [CrossRef]
14. Schmidt, C.G.; Wagner, S.M. Blockchain and supply chain relations: A transaction cost theory perspective. *J. Purch. Supply Manag.* **2019**, *25*, 100552. [CrossRef]
15. Androulaki, E.; Barger, A.; Bortnikov, V. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference 2018, Porto, Portugal, 23–26 April 2018.

16. Hyperledger Blockchain Performance Metrics White Paper—Hyperledger. Available online: https://www.hyperledger.org/learn/publications/blockchain-performance-metrics# (accessed on 20 September 2021).
17. Valenta, M.; Sandner, P. Comparison of Ethereum, Hyperledger Fabric and Corda. *Ebook Frankf. Sch. Blockchain Cent.* **2017**, 1–8. Available online: http://www.smallake.kr/wp-content/uploads/2017/07/2017_Comparison-of-Ethereum-Hyperledger-Corda.pdf (accessed on 23 October 2021).
18. Nasir, Q.; Qasse, I.A.; Abu Talib, M. Performance analysis of hyperledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]
19. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
20. Vivar, A.L.; Orozco, A.L.S.; Villalba, L.J.G. A security framework for Ethereum smart contracts. *Comput. Commun.* **2021**, *172*, 119–129. [CrossRef]
21. Chen, C.L.; Deng, Y.Y.; Weng, W.; Sun, H.; Zhou, M. A Blockchain-Based Secure Inter-Hospital EMR Sharing System. *Appl. Sci.* **2020**, *10*, 4958. [CrossRef]
22. Yogesh, P.R. Formal verification of secure evidence collection protocol using BAN logic and AVISPA. *Procedia Comput. Sci.* **2020**, *167*, 1334–1344. [CrossRef]