International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php

Ensuring Privacy In Controlling Access And Sharing Of Information By Supply Chain Entities

Noor Hadi Hammood [1*], Amir Jalaly Bidgoly [1]

[1] Department of Computer and IT, University of Qom, Qom, Iran
[*] Corresponding author email (noor.h.aljanabi@gmail.com)

Abstract
The importance of using supply chains in industries cannot be denied. Supply chain and its management are vital in any organization and involve coordinating all activities related to producing and delivering goods and services to the end consumer. This complex process requires effective planning, execution, and monitoring. Effective supply chain management ensures that a company can minimize costs, increase efficiency, and maximize profits. In this paper, the concepts and dimensions of supply chain security are examined, and then the effects of cyber threats, fraud, and natural disasters on the supply chain are discussed. After studying the dimensions of security and privacy in the supply chain, we have concluded that the current research has not examined the secure sharing of information in the supply chain. For this reason, our primary focus in this paper is on data confidentiality and the privacy of participants' information. In this paper, we have presented a comprehensive protocol for a supply chain that considers the aspects of information storage, access control, authentication, key agreement, and data sharing in the supply chain. Next, we evaluated our proposed scheme informally and formally using the Scyther tool, and the results obtained indicate that the proposed scheme is secure against known attacks and also meets security requirements. Finally, we show that the proposed scheme performed well in execution time.

Keywords: *Information security, confidentiality, privacy, authentication, supply chain, information sharing.*

1. INTRODUCTION

With the rapid progression of scientific and technological developments across diverse industrial domains and everyday life, there has been a pronounced shift among enterprises, manufacturers, and production facilities toward incorporating automation and innovation into their product offerings. In pursuit of enhanced competitiveness, many have strategically adopted supply chain frameworks to strengthen their market position. When effectively implemented and managed, supply chain systems offer a structured approach to optimizing production processes and can significantly contribute to improved operational outcomes and increased profitability [1].

The supply chain includes a set of processes to provide a product or service to the consumer. These processes include converting raw materials into finished products and transporting and distributing these products to end users. Participants in the supply chain include manufacturers, sellers, storage facilities, logistics companies, distribution centers, and retailers [2]. A supply chain's components include all tasks from receiving an order to meeting customer needs. These tasks include product development, marketing, operations, distribution channels, financial aspects, and customer support [3].

From a top-down perspective and a macro view of supply chain components, supply chain components can be divided into three parts: upstream, downstream, and internal.

- **Downstream supply chain:** The Downstream supply chain includes primary suppliers and raw material producers. Supplier relationships can develop at various levels up to the first raw material supplier. In this part, the main activity is procurement.
- **Internal supply chain:** This part includes all the activities that convert the data brought into the organization into outputs. These activities include material transportation, inventory management, manufacturing, and quality control. This chain must continue until the product moves outside the organization for distribution.
- **Upstream supply chain:** This stage includes the processes of distributing and delivering the product to the final customers. The activity of this chain includes packaging, transportation, and warehousing. This chain ends when the product is transferred or consumed.

Of course, it is also necessary to mention that, as can be seen in Figure 1, the internal chain itself can play the role of upstream chain for downstream chain and the role of downstream chain for upstream chain.

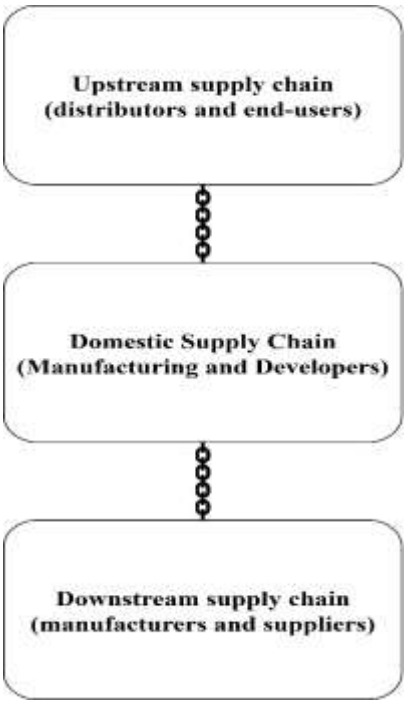

Figure 1. A macro top-down view of the supply

While supply chains have significantly enhanced operational efficiency and value generation across organizational and individual levels, insufficient prioritization of their security dimensions particularly concerning data confidentiality and stakeholder privacy remains a persistent and unresolved issue within the discipline. As the proliferation of big data reshapes supply chain management, safeguarding information becomes increasingly critical. The datasets exchanged across supply networks often encompass proprietary and sensitive content relating to customers, suppliers, products, workflows, and performance metrics [4]. These datasets are instrumental for deriving actionable insights, streamlining logistical operations, and driving stakeholder value. Nonetheless, they are also susceptible to a range of security threats, including cyber intrusions, data leaks, unauthorized disclosures, and malicious exploitation [5]. To mitigate these risks, it is imperative that supply chain professionals implement robust data protection strategies and uphold the privacy rights of all actors involved—from upstream raw material providers to downstream end users—throughout the entire data exchange continuum [6] [7].

This study aims to address a critical and unresolved challenge in supply chain systems: ensuring the security and confidentiality of data produced throughout the supply chain, as well as safeguarding the privacy of all participating entities. To this end, we introduce a security protocol architecture tailored to various operational stages of the supply chain, including data storage, access control, and information exchange. The proposed framework integrates established cryptographic and security mechanisms to preserve both the confidentiality of sensitive information and the privacy of distinct stakeholders involved in the process. This approach facilitates secure and reliable supply chain management while maintaining operational efficiency.

Section 2 provides a survey of related research, followed by a detailed presentation of the proposed protocol in Section 3. Section 4 offers a security analysis of the system, while Section 5 assesses its performance and computational efficiency. Concluding remarks and directions for future investigation are presented in Section 6.

2. REVIEW OF PREVIOUS WORK

In this section, we will review the research conducted on supply chain security, considering the purpose of the problem, and we will review the plans that researchers have presented to improve supply chain security one by one.

Warren and Hutchinson [8] first stated the main security risks related to supply chain and supply chain management in their proposed scheme. According to them, the main attacks in that year were password guessing attacks, impersonation attacks, denial of service attacks, and others.

Zhang et al. [9] also focused on possible security attacks that may occur in the supply chain during information sharing. In this regard, their research addressed the possible threats that may occur during information sharing in the supply chain. Although Zhang et al.'s research contributed greatly to the recognition of known attacks, due to the emerging nature of the supply chain and its security, the threats and security requirements were not fully known at the time of their research.

Sindhuja and Kunnathur [10] developed information sharing architectures with appropriate design frameworks, scalability, and consideration of different security levels and participant requirements. These architectures included security services such as authentication, delegation, access control, data confidentiality and integrity, auditing, and non-repudiation. Four security levels were defined based on the importance of the data, and security needs and threats were identified for each level. Finally, an appropriate security service was formed. The use of information technology and computer science in the supply chain increases the efficiency of the supply chain. However, despite all the benefits of using information technology and computer science in the supply chain, it undoubtedly increases security and cyber challenges in this area. Balancing data security, participant privacy, and supply chain efficiency is very important. For this reason, Smith et al. [11] presented a model to balance the efficiency that information technology can bring to the supply chain and the security challenges it may create for the supply chain. Smith et al. aimed to use the efficiency of information technology alongside the security challenges it may bring to supply chain management.

To address the problem of inference-based information leakage, Zhang et al. [12] introduced a conceptual framework aimed at enhancing supply chain stakeholders' understanding of this specific threat. Alongside the model, they proposed a quantitative methodology for evaluating the risk associated with such leakage, thereby enabling organizations to identify sensitive information and implement appropriate risk assessment and mitigation strategies. In a related contribution, Lin et al. [13] developed a cloud-oriented architecture for secure data storage and exchange within the supply chain. Their framework incorporates authentication mechanisms and protocols for ownership transfer, wherein data is encrypted using the owner's cryptographic key, and access rights are mediated through a trusted third party. This design enhances supply chain transparency while asserting resistance to several known security threats. Nevertheless, Khor and Sidorov [14], through critical examination in 2018, demonstrated that the proposed protocol was susceptible to denial-of-service (DoS) and out-of-synchronization attacks. They subsequently introduced enhancements to reinforce the protocol's resilience against such vulnerabilities.

Xiaoming [15] proposed a blockchain scheme for sharing information in the supply chain, which combined it with homomorphic encryption. In blockchain, the user owns their encrypted data and knows what data is collected for their purpose and how it is used. However, Guipeng et al. [16] proved in 2023 that the scheme proposed by [15] is vulnerable.

Preuveneers et al. [17] proposed a distributed trust and decentralized access control model based on a private blockchain in 2017 to provide transparency, integrity, authenticity, and permissioned data flow in Industry 4.0. The challenge of this scheme is related to the privacy and scalability of private blockchain. This paper protects unauthorized adversary access to systems, services, and information. It cannot prevent identification and link ability attacks, and as private blockchains scale up, it becomes difficult for resource-constrained devices to mine (or verify) all transactions.

Gao et al. [18] integrated blockchain and supply chain and tried to overcome the limitations of using blockchain for data protection. Gao et al. used AES encryption to protect supply chain information in their proposed method. They assumed that the communication between different supply chain components is secure and that the attacker cannot eavesdrop or tamper with the exchanged messages. It also assumed the trust and honesty of some participants in the supply chain. In general, it reduced the complexity and improved the throughput, but its performance was low and unsuitable for supply chain designs with different models.

Sidorov et al. [19] proposed a lightweight mutual authentication protocol that uses a blockchain network with access control capabilities to provide security in a decentralized supply chain. However, this protocol does not provide sufficient security measures, as an adversary can easily obtain sensitive parameters exchanged between the parties. After reviewing and analyzing the design of [19] , Jangirala et al. [20] proposed a secure authentication protocol called LBRAPS for supply chains in edge computing environments. Their proposed protocol primarily addressed the security needs, but the computational cost of their proposed protocol was high.

Kumar et al. [21] proposed a very lightweight RFID authentication protocol based on blockchain technology for supply chains. Their aim was to provide a scheme that could strike a balance between security and efficiency. However, the proposed scheme could not consider all the security requirements and was not suitable for supply chains.

Dwivedi et al. [22] introduced a blockchain-driven framework in 2020 to facilitate secure information exchange within the pharmaceutical supply chain, leveraging smart contracts and consensus mechanisms. Their design also utilized smart contracts for the administration of encryption keys. The accompanying security evaluation demonstrated the protocol's robustness against

various attack vectors, while also indicating that the communication and computational overhead remained within efficient bounds. Nonetheless, the proposed scheme was noted for incurring substantial operational overhead.

Research by Abidi et al. [6] proposed a scheme for preserving privacy in blockchain-based supply chains. The method consisted of three steps. In the first step, sensitive data is identified. In the second step, this data is encrypted with a secret key and sent to the recipient through the blockchain network. Finally, the recipient accesses the data using a recovery or decryption operation.

In 2021, Bader et al. [23] proposed a reliable and scalable blockchain infrastructure to improve information availability and accountability in large supply chains. In fact, Bader et al.'s goal was to ensure that the privacy of participants is not violated and that the supply chain operates properly while maintaining the requirement for transparency.

Liu et al. [24] proposed an attribute-based access control to address privacy and anonymity leakage in blockchain-based supply chains and increase security, transparency, and availability. This scheme provides data confidentiality and strict access control by multiple authority managers for secure information sharing in the supply chain. However, the proposed scheme by Liu et al. used heavy operators to design their proposed scheme, making it unsuitable for supply chain infrastructure and the systems in which it is used.

Gómez-Marín et al. [25] used security hardware modules to authenticate and verify assets in the supply chain. In this method, the private key of the components is stored in the SE (Trusted Element) to provide security and integrity. Transactions made on the chain are signed to ensure ownership and authenticity. It also uses encryption to protect sensitive supply chain information. The proposed method resists attacks such as forgery, beneficiary denial, and misleading information. However, this scheme required a trusted entity to be considered, and its proposed scheme also used heavyweight operators. It did not consider the energy, memory, and computational limitations of the devices available to participants in the supply chain.

3. PROPOSED SCHEME

In the following section, we will describe our proposed protocol. As mentioned, our proposed protocol consists of three main components: access control, storage, and sharing. As seen in Figure 2, at the beginning, each downstream entity must register with its upstream entity and, to share information, perform two-way authentication. Then it will be allowed to store the information. Suppose the information requester in the supply chain intends to access the information. After being authenticated for Mo, each downstream entity can view additional information on other entities in the supply chain after being authenticated for the upstream entity and send an access request for each. Table 1 shows the symbols used in the proposed protocol.

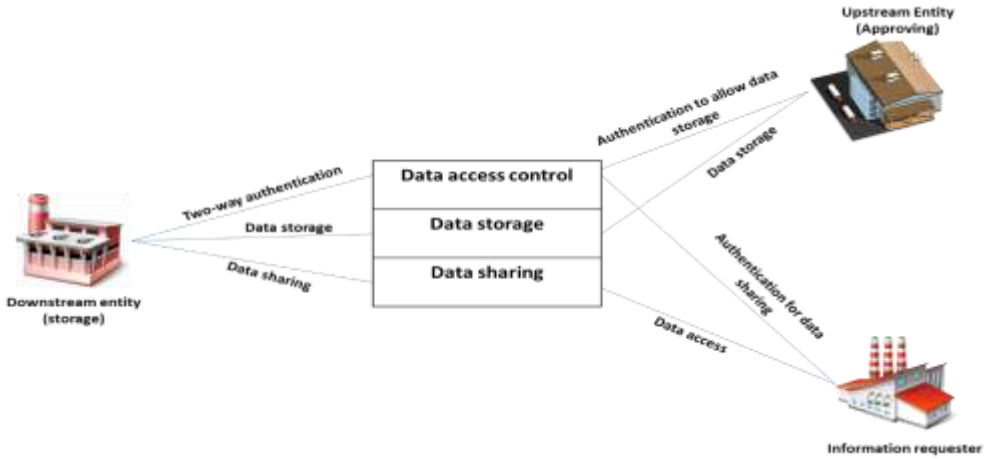

Figure 2: General scheme of the proposed scheme

Table 1: Symbols used in the proposed protocol

Symbol	Description
U_m	mth user (data owner)
ID_m	mth user id
ID_b	Upstream user ID in the supply chain
PW_m	User's chosen password m
f	Data file
$Sign_f$	Digital signature for file f
a_f	Meta data
U_j	Information requester
ID_j	Information requester ID
p	Elliptic curve cryptographic generator
$PR_b, PU_b = PR_b.p$	Private key/public key of the upstream entity
$PR_m, PU_m = PR_m.p$	Data owner private key/public key
$PR_j, PU_j = PR_j.p$	Private key/public key of the information requester
sk_i, sk_j	Session key
, ΔT ⊕	XOR operator and maximum transmission delay
h(.)	hash
ENC_k ()/DEC_k ()	Encryption and decryption
| |	Two-string concatenation operator

3.1. Registration stage of the downstream entity in the upstream entity of the supply chain

In this section, we will explain the registration phase of the proposed protocol. For a chain to be formed, each downstream entity must register with the upstream entity in the supply chain, and the upstream entity must register with its upstream entity to form the desired chain.

At the beginning of the work, the downstream entity selects the identifier ID_m, the password pw_m, and two random numbrs q_m and r_m for itself, calculates the parameter A_i according to the relation (1), and then sends the identifier ID_m and the parameter A_i to the upstream entity. As soon as the upstream entity receives the parameters sent from the downstream entity, it calculates the parameters B_i and C_i according to the relations (2) and (3), and finally stores the parameters ID_b , B_i, and C_i inside the mobile device or sends them securely to the downstream entity through a secure channel as can be seen in Figure 3. The downstream entity also adds two random numbers, q_m and r_m, to the mobile phone memory, and finally, the registration phase ends. It is important to note that the communication channel in the registration phase is considered completely secure, and we also consider the communication channel insecure in the authentication and key agreement phase.

$$A_i = h(pw_m||r_m)\oplus q_m \quad (1)$$
$$B_i = h(ID_b \oplus PR_b) \quad (2)$$
$$C_i = A_i \oplus B_i \quad (3)$$

Downstream entity (data owner)	Upstream entity
Chooses ID_m, pw_m Select random numbers r_m, q_m Calculates $A_i = h(pw_m\|\|r_m)\oplus q_m$ $\{ID_m, A_i\}$ →	
	Computes $B_i = h(ID_b \oplus PR_b)$ $C_i = A_i \oplus B_i$ Stores B_i , C_i , ID_b into the mobile device ← mobile device
Saves r_m, q_m in to the mobile device	

Figure 3: Registration stage of the proposed scheme

3.2. Authentication and key agreement stage between entities in the supply chain

In this section, if one of the downstream entities intends to store or send data to its upstream entity in the supply chain, authentication must be established between the upstream entity and the downstream entity at the beginning of the work to determine that the downstream entity is the one it claims to be and has passed the registration stage.

At the beginning of the authentication and key agreement stage, it must be determined that the mobile phone is in the possession of its owner, so as can be seen in Figure 4, the user enters the ID_m^* and the password pw_m^* selected in the registration stage into his mobile phone. Next, it must be determined whether the entered ID and password are correct or not, so relations (4) and (5) are calculated, and finally the calculated parameter C_i^* is compared with the parameter C_i that is in the mobile phone memory, as can be seen in relation (6). If these two parameters are equal, it is determined that the person entering the password is the owner of the mobile device, and the login step has been completed.

$$A_i^* = \mathrm{h}(pw_m^* || r_m) \oplus q_m \quad (4)$$

$$C_i^* = A_i^* \oplus B_i^* \quad (5)$$

$$C_i^* ? = C_i \quad (6)$$

Next, the timestamp T_1 is selected by the downstream entity, or rather, the mobile phone in the downstream entity's possession. It also selects a random number y_i, then calculates the parameters U_i and Q_i based on relations (7) and (8) and sends a message containing the parameters C_i, T_1, Q_i , and U_i to the upstream entity.

$$U_i = h(y_i || T_1 || C_i || ID_b) \quad (7)$$

$$Q_i = y_i \oplus C_i \oplus A_i \quad (8)$$

When the upstream entity receives a message sent from the downstream entity, it initially checks the freshness of the message by checking equation (9). If the freshness of the message is proven, it calculates the parameters B_i^*, A_i^*, y_i^*, and U_i^* according to equations (10) to (13). It compares the calculated parameter U_i^* with the parameter U_i sent from the downstream entity. If they are equal, the accuracy of the information sent and the identity of the sender of the message are verified.

$$\Delta T >= |T_2 - T_1| \quad (9)$$

$$B_i^* = h(ID_b \oplus PR_b) \quad (10)$$

$$A_i^* = C_i \oplus B_i^* \quad (11)$$

$$y_i^* = Q_i \oplus C_i \oplus A_i^* \quad (12)$$

$$U_i^* = h(y_i^* || T_1 || C_i || ID_b) \quad (13)$$

Next, the upstream entity selects a random number x_i and, according to relations (14) to (17), calculates the parameters z_i, W_i, sk_i , which is the key agreed upon between the parties, and $Auth_i$. Finally, it sends a message to the downstream entity that includes the parameters W_i, the timestamp T_2, and the parameter $Auth_i$.

$$\Delta T >= |T_2 - T_1| \quad (14)$$

$$B_i^* = h(ID_b \oplus PR_b) \quad (15)$$

$$A_i^* = C_i \oplus B_i^* \quad (16)$$

$$y_i^* = Q_i \oplus C_i \oplus A_i^* \quad (17)$$

When the downstream entity receives the message sent from the upstream entity in the supply chain, it checks the freshness of the received message based on equation (18). Next, in order to calculate the agreed key and also check the authenticity of the sent message and the identity of the sender, it calculates the parameters z_i^*, x_i^*The agreed key sk_i^* and the parameter $Auth_i^*$ based on equations (19) to (22). Next, as can be seen in Figure 4, it compares the parameter $Auth_i^*$ with the parameter $Auth_i$ sent from the upstream entity. If these two parameters are equal, the identity of the sender of the message, i.e., the upstream entity, as well as the authenticity or integrity of the sent message, is also proven for the downstream entity.

$$\Delta T >= |T_3 - T_2| \quad (18)$$

$$z_i^* = W_i \oplus A_i^* \oplus B_i \quad (19)$$

$$x_i^* = z_i^* \oplus y_i \quad (20)$$

$$sk_i^* = h(x_i^* || y_i || T_2 || z_i^* || T_2) \quad (21)$$

$$Auth_i^* = h(x_i^* || W_i || T_2 || sk_i^*) \quad (22)$$

As clarified in this section, the two upstream and downstream entities agreed on a session key after mutual authentication. The following considers another step called password change, which allows the downstream entity to change the password selected during the registration stage if it wants to do so using the method mentioned.

Downstream entity (data owner)	Upstream entity
Inputs ID_m^* and pw_m^* A_i^* =h($pw_m^*||r_m$)⊕q_m $C_i^* = A_i^* \oplus B_i^*$ $C_i^* ? = C_i$ Select time stamp T_1 Select random number y_i $U_i = h(y_i \| \| T_1 \| \| C_i \| \| ID_b)$ $Q_i = y_i \oplus C_i \oplus A_i$ $\{C_i, T_1, Q_i, U_i\}$ → Public channel	
	Select time stamp T_2 ΔT>=$\|T_2 - T_1\|$ Computes $B_i^* = h(ID_b \oplus PR_b)$ $A_i^* = C_i \oplus B_i^*$ $y_i^* = Q_i \oplus C_i \oplus A_i^*$ $U_i^* = h(y_i^* \| \| T_1 \| \| C_i \| \| ID_b)$ Verifies $U_i^* ? = U_i$ Select random number x_i Computes $z_i = x_i \oplus y_i$ $W_i = z_i \oplus A_i \oplus B_i$ $sk_i = h(x_i \| \| y_i \| \| T_1 \| \| z_i \| \| T_2 \| \| B_i)$ $Auth_i = h(x_i \| \| W_i \| \| T_2 \| \| sk_i)$ $\{W_i, T_2, Auth_i\}$ ← Public channel
Select time stamp T_3 ΔT>=$\|T_3 - T_2\|$ Calculates $z_i^* = W_i \oplus A_i^* \oplus B_i$ $x_i^* = z_i^* \oplus y_i$ $sk_i^* = h(x_i^* \| \| y_i \| \| T_1 \| \| z_i^* \| \| T_2 \| \| B_i)$ $Auth_i^* = h(x_i^* \| \| W_i \| \| T_2 \| \| sk_i^*)$ Checks $Auth_i^* = ? Auth_i$	

Figure 4: Authentication and key agreement stage of the proposed scheme

3.3. Password change step

The purpose of designing this step is that the entity with the mobile device in the registration step, if it intends to change the password selected in the registration step, can change its password at this step and replace it with a new one. In this step, it must be determined at the beginning that the mobile phone is in the possession of its owner and has not been stolen; for this reason, it first enters the identifier ID_m^* and the password pw_m^*. Then, according to relations (23) to (24), it calculates the parameters A_i^* and C_i^* and checks the equality of the parameters C_i^* and C_i . If they are equal, it is determined that the owner of the mobile phone has entered the identifier and password and the login step is completed.

$$A_i^* = \mathrm{h}(pw_m^* || r_m) \oplus q_m \quad (23)$$

$$C_i^* = A_i^* \oplus B_i^* \quad (24)$$

Next, the downstream entity enters a password change request and enters its new password, pw_m^{**}. Next, the upstream entity calculates relations (25) to (26), and the parameter C_i^{**} replaces C_i^* in the mobile phone.

$$A_i^{**} = \mathrm{h}(pw_m^* || r_m) \oplus q_m \quad (25)$$

$$C_i^{**} = A_i^{**} \oplus B_i^* \quad (26)$$

3.4. Data storage stage

As mentioned, the goal of this step is to describe how information is stored in the memory of entities and the parameters needed to store it. In our proposed scenario in the supply chain, each downstream entity must register with its upstream entity and also store its parameters and information about the products it produces or exchanges with its upstream entity. The upstream entity also stores its information in its upstream entity, and so on until a chain is formed.

However, the storage method is critical and must be such that, in the future, if one of the other entities in the supply chain or a third party can request access to this information.

At this stage, assume that U_m is the downstream entity or the data owner that authenticated and agreed on the key with the upstream entity in the previous stage. It was determined by the upstream entity that it is the one who claims to own the data and intends to store its data in the upstream entity. The data that the downstream entity intends to store is divided into confidential data and metadata, as mentioned above. Metadata is information about confidential data that other entities can view if they want to access the confidential data and send their request to the data owner if they want to access this data. To perform the storage stage, the entity U_m performs the following steps to store its data file, i.e. f, in the upstream entity in the supply chain.

At the beginning of the work, U_m takes a hash of its data file according to equation (27), or in other words, applies the hashing function to its file to obtain the parameter hk. Then, it selects a random number w_i and encrypts the original data file along with the random number w_i with the data hash, i.e. the parameter hk, according to equation (28).

$$\mathrm{hk}=\mathrm{h}(\mathrm{f}) \quad (27)$$

$$c_f = ENC_{hk}(f || w_i) \quad (28)$$

Next, the entity U_m selects a random number m_i and calculates the parameters M_i and $D_f$according to relations (29) to (30).

$$M_i = m_i . p \quad (29)$$

$$D_f = c_f . p \quad (30)$$

$$sign_f = ENC_{PR_m}(c_f || D_f) \quad (31)$$

Next, the entity U_m selects a random number l_i and, as you can see in equations (32) to (34), calculates the parameters L_i, k_b , and K_m. Next, the user will encrypt the parameters ID_m and l_i with the parameter K_m according to equation (35).

$$L_i = l_i . p \quad (32)$$

$$k_b = l_i . PU_b \quad (33)$$

$$K_m = h(c_f || ID_m) \quad (43)$$

$$CF_i = ENC_{K_m}(ID_m || l_i) \quad (35)$$

Finally, entity U_m will send parameters c_f, K_m, CF_i, $sign_f$ and L_i to the upstream entity in the supply chain. The upstream entity first checks the downstream entity's signature on the data based on equation (36) and then signs parameter c_f with its own private key according to equation (37) to obtain parameter ST_i and finally parameters ST_i, $sign_f$ and ID_m which are the identifier of the data owner entity or the downstream entity will be stored in the memory of the upstream entity which can be a cloud service provider or blockchain.

$$Verify(sign_f) = DEC_{PU_m}(c_f || D_f) \quad (36)$$

$$ST_i = ENC_{PR_b}(c_f) \quad (37)$$

3.5. Data sharing stage in the supply chain

At this stage, as mentioned in the explanation, the goal is to allow another entity in the supply chain to access this entity's sensitive data after seeing its metadata.

At this stage, the entity U_j, or the information requester from one of the supply chain entities, can search the existing metadata a_f in the supply chain entities. Next, it requests the original files related to the metadata. In that case, it must send a request for access and sharing of the original data to the upstream entity so that if the data owner allows access to the data, the original data can be shared with them.

When the information requester $U_j$searches the metadata, they request access to a file. The upstream entity checks the access request $U_j$and if information is available for the request U_j, it first calculates the parameter k_b' according to equation (38). Then it sends the entity identifier $U_j$to the data owner in encrypted form based on equation (39) to the entity U_m, the owner of the data file.

$$k_b' = PR_b.L_i \tag{38}$$
$$EN_j = ENC_{k_b'}(ID_j) \tag{39}$$

Given that the data owner has the parameter k_b and also considering that the parameter k_b is equal to the parameter k_b', the data owner will decode the parameter EN_j according to equation (40) and obtain the identifier of the information requester, i.e. ID_j.

$$ID_j = DEC_{k_b}(EN_j) \tag{40}$$

The data owner in the supply chain, or the downstream entity in the supply chain, will encrypt the parameter k, which is actually the encryption key of the data file, according to equation (41), using the public key of the data requester, and send the information to the requester.

$$KF_i = ENC_{PU_j}(k) \tag{41}$$

Now the data requester downloads the encrypted files from the upstream entity database, which as mentioned can be a blockchain or a cloud service provider. The file is encrypted and the data requester must be able to obtain the decryption key k using the parameters sent to him by the data owner U_i. In this regard, the data requester U_j then performs the following operation (42) to obtain the parameter k.

$$k=DEC_{PU_m}(KF_i) \tag{42}$$

Now, considering that the data requester has the data file and the file key, the user U_j will decrypt the parameter c_f using the parameter k according to equation (43) and obtain the original data file, i.e. f.

$$F||w_i=DEC_k(c_f) \tag{43}$$

Finally, entity U_j has the ability to take a hash from the data file and compare it with parameter k. If these two parameters are equal, it will be determined that the integrity of the original file has been maintained and the data file has reached the data requester in its correctness and integrity.

4. Proof of the security of the proposed scheme

In this section, we will prove the proposed scheme from a security perspective in two ways: formally using the Scyther tool [26] and informally using proof and argument. We will show that the proposed scheme can meet various security requirements and is resistant to known attacks. Next, we will informally prove the proposed scheme using proof and argument.

4.1. Anonymity

Under the anonymity security requirement, it is assumed that if attackers intercept the communication channel, they will not be able to determine the identity of the message sender and receiver. In the proposed protocol, the identifiers of any of the entities involved in the communication, namely ID_b and ID_m, are not sent over the communication channel, which is why we claim that our proposed scheme meets this security requirement.

a) Confidentiality

The security requirement of confidentiality means that attackers cannot access sensitive parameters exchanged by the entities involved in the protocol. However, in the proposed protocol, sensitive parameters such as the password pw_m, the identifiers of the parties, namely ID_b and ID_m, or the session key SK_m, or any other parameter, are not exchanged on the public channel during the authentication and key agreement phase. Therefore, it can be claimed that the security requirement of confidentiality is maintained in the proposed protocol.

b) Perfect Forward Secrecy (PFS)

Another sensitive and well-known security requirement is the security requirement of perfect forward secrecy. In this security requirement, it is assumed that if an attacker can obtain sensitive parameters such as the parties' private keys, he should not be able to obtain the session key agreed between the parties. In the proposed protocol, in the session key $sk_i = h(x_i||y_i||T_2||z_i||T_2||B_i)$ there

are parameters x_i and y_i which are random numbers generated during the protocol steps. Therefore, even if attackers can obtain sensitive parameters such as the parties' private keys, they will still not be able to obtain and reconstruct the session key agreed between the parties.

c) Mutual authentication

The access control component's primary purpose was to authenticate the parties. In the proposed protocol, as explained above, authentication between the parties is performed in two stages. As is clear in the authentication and key agreement stage, when the equality $U_i^* ? = U_i$ is checked the downstream entity is authenticated for the upstream entity. Also, if the parameters $Auth_i^* = ? Auth_i$ are equal, the upstream entity is authenticated for the downstream entity in the supply chain. Therefore, our proposed scheme meets this security requirement.

d) known session-specific temporary information security

In this attack, it is assumed that if the attacker can obtain the random parameters generated during the protocol, he should not be able to reproduce the session key. However, in the proposed protocol, the session key $sk_i = h(x_i||y_i||T_2||z_i||T_2||B_i)$ includes the parameter B_i , which is dependent on the secret key of the upstream entity and also its ID. As a result, even if the attacker obtains the random parameters, he cannot obtain the parameter B_i . Therefore, it is resistant to the attack of revealing random parameters.

e) Stolen Verifier attack (SV attack)

In this attack, it is assumed that if the attacker can access the parameters stored in the memory of the parties, he should not be able to access the agreed session key. However, in the session key $sk_i = h(x_i||y_i||T_2||z_i||T_2||B_i)$ there are parameters x_i and y_i which are random numbers and are generated during the protocol steps, so even if the attacker accesses the memory of the communicating parties, he will still not be able to generate the session key.

f) Insider attack

In this attack, as explained in the Basic Concepts chapter, the goal is to prevent the user's chosen password from being disclosed to the upstream entity during the registration stage so that there is no possibility of misuse. For this purpose, in the proposed protocol, the downstream entity's chosen password is sent to the upstream entity in the format $A_i = h(pw_m||r_m) \oplus q_m$. For this reason, even if there is an attacker on the upstream entity side, the downstream entity's password is not disclosed to him, so there is no possibility of misuse.

g) Replay attack

In this attack, the goal is for the attacker to resend stale and repetitive messages to the parties. In the proposed protocol, time stamps are used to prevent this attack. When each message is received, the freshness of the received message is checked, and if the desired message is not fresh, it is discarded.

h) Impersonation attack

Another well-known attack is the impersonation attack. In this attack, the attacker tries to impersonate one of the legitimate entities of the protocol by manipulating the messages exchanged on the public channel and communicating with other entities. However, in our proposed protocol, we perform mutual authentication by checking $U_i^* ? = U_i$ and $Auth_i^* = ? Auth_i$ so attackers will not be able to perform the impersonation attack.

i) Denial-of-service (DoS) attack

In the proposed protocol, because the freshness of the messages received by the parties is checked using time stamps, the attacker cannot send messages repeatedly and consecutively to the parties, which are legal entities that cannot respond to or service. Due to the use of time stamps and checking the freshness of the messages, such an attack is not possible.

4.2. Formal security proof of the proposed scheme using the Scyther tool

Scyther is a powerful and effective tool for analyzing and identifying potential attacks and vulnerabilities in security protocols. It is used in security protocols, especially for analyzing and synthesizing security features. This program is designed to help researchers and cybersecurity professionals verify the correctness of security protocols and identify potential vulnerabilities. Below, you can see the pseudocode (Figure 5) and output of the Scyther tool for the proposed protocol.

Figure 6 shows the output of the proposed protocol check by Scyther, and Figure 7 shows the coding platform of the Scyther tool. The Niagree feature ensures that the communicating parties are reliably aware of the secure and orderly transmission of messages between them. The Nisynch feature ensures that the messages exchanged between the parties cannot be decrypted or retransmitted. The Alive feature confirms that the communicating parties have confirmed the sequence of protocol steps. The Weakagree feature ensures no possibility of spoofing in the protocol. In addition, the secret attribute ensures that the corresponding parameter remains secure.

Pseudocode:

1. Initialization
Define cryptographic functions:
  h(), XOR()
Secrets:
  idm, pwm, pwmm, rm, qm, idb, prb
Compute helper values:
  Ai = XOR(h(pwm, rm), qm)
  Bi = h(XOR(idb, prb))
  Ci = XOR(Ai, Bi)

2. Role: L-SupplyChain
Compute temporary values:
  Aii = XOR(h(pwmm, rm), qm)
  Bi = h(XOR(idb, prb))
  Cii = XOR(Aii, Bi)
Verify:
  if Ci != Cii → Abort
Generate fresh nonce:
  yi
Compute:
  Ui = h(yi, Ci, idb)
  Qi = XOR(yi, Ci, Ai)
Send:
  (Ci, Qi, Ui) → H-SupplyChain
Receive:
  (Wi, Authi) ← H-SupplyChain
Compute:
  zii = XOR(Wi, Aii, Bii)
  xii = XOR(zii, yi)
  skii = h(xii, yi, zii, Bi)
  Authii = h(xii, Wi, skii)
Verify:
  if Authii != Authi → Abort

3. Role: H-SupplyChain
Receive:
  (Ci, Qi, Ui) ← L-SupplyChain
Compute:
  Bii = h(XOR(idb, prb))
  Aii = XOR(Ci, Bii)
  yii = XOR(Qi, Ci, Aii)
  Uii = h(yii, Ci, idb)
Verify:
  if Uii != Ui → Abort
Generate fresh nonce:
  xi
Compute:
  zi = XOR(xi, yi)
  Wi = XOR(zi, Ai, Bi)
  ski = h(xi, yi, zi, Bi)
  Authi = h(xi, Wi, ski)

Send:
  (Wi, Authi) → L-SupplyChain

Figure 5: Proposed Supply-Chain-Protocol in Pseudocode

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php

Scyther results : autoverify

Claim				Status	Comments
Supply_Chain_Protocol	L_SupplyChain	Supply_Chain_Protocol,L_SupplyChain1	Secret _Hidden_ 2	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain2	Secret yi	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain3	Secret _Hidden_ 1	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain4	Secret Bii	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain5	Secret Wi	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain6	Secret Authi	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain7	Alive	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain8	Weakagree	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain9	Niagree	Ok	No attacks within bounds.
		Supply_Chain_Protocol,L_SupplyChain10	Nisynch	Ok	No attacks within bounds.
	H_SupplyChain	Supply_Chain_Protocol,H_SupplyChain1	Secret xi	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain2	Secret _Hidden_ 3	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain3	Secret yi	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain4	Alive	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain5	Weakagree	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain6	Niagree	Ok	No attacks within bounds.
		Supply_Chain_Protocol,H_SupplyChain7	Nisynch	Ok	No attacks within bounds.

Done.

Figure 6: Scyther tool output for the proposed protocol

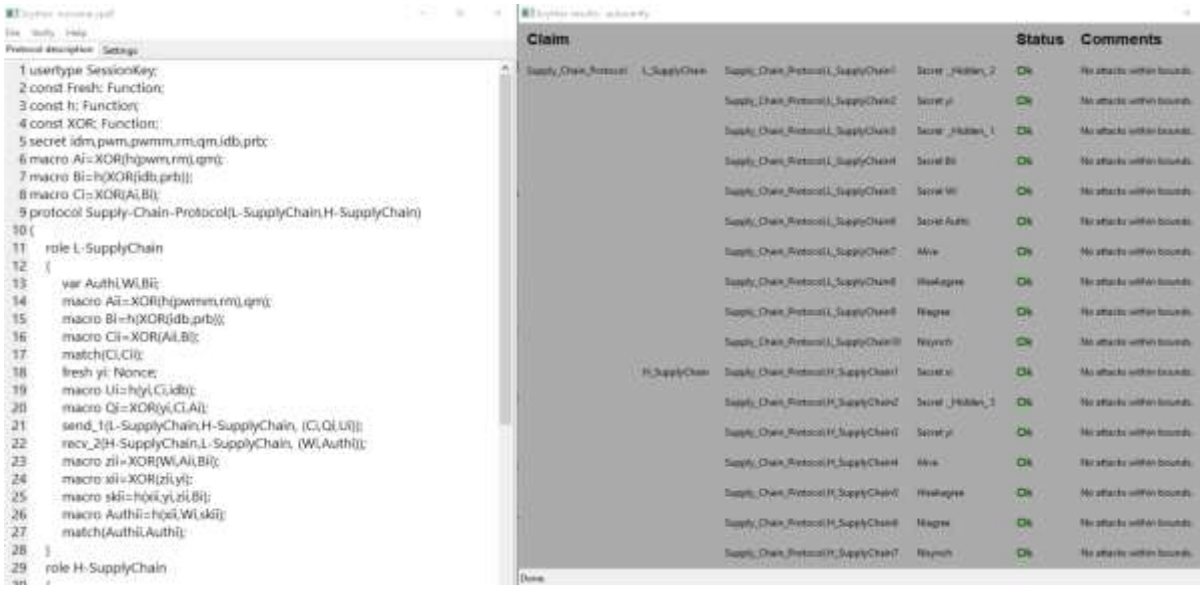

Figure 7: Scyther tool coding platform with tool output

As stated in this section, the proposed scheme will be able to meet various security needs and resist known attacks. In this regard, it was stated both formally and informally that the proposed scheme will be able to meet the security needs of perfect forward

secrecy, anonymity, confidentiality, and mutual authentication. The proposed scheme will also be resistant to known attacks such as impersonation, replay attack, insider attack, authenticator theft attack, random parameter disclosure attack, and denial of service attack.

In Table 2, we have compared the proposed scheme with similar schemes in terms of security and shown that the proposed scheme also performs better than similar schemes in terms of security. It is worth noting that the research that has considered the storage and sharing stage in its proposed scheme is minimal, and there is no comprehensive scheme in the supply chain that can comprehensively consider the information storage and sharing stage in its proposed scheme

Table 2: Security comparison of the proposed scheme with other similar schemes

Ref. No.	[28]	[22]	[27]	Proposed scheme
anonymity	×	~	✓	✓
perfect forward secrecy	✓	~	×	✓
known session-specific temporary information attack	✓	×	~	✓
stolen verifier attack	×	~	~	✓
replay attack	✓	✓	✓	✓
Insider attack	✓	~	✓	✓
Dos attack	✓	✓	✓	✓
impersonation attack	×	✓	✓	✓
Formal proof using security tools	×	×	✓	✓
Considering the storage and sharing phase	~	✓	~	✓

✓ Feature confirmation | ×Lack of supply or resistance | ~No review

5. Check the efficiency of the proposed scheme

In this section, we will calculate the proposed scheme's execution time. In fact, similar schemes measure the time of protocol schemes by calculating the specific operators used in the protocol in terms of the number and duration of execution of each operator and calculating the overall time of the protocol.

In this study, according to the times measured and presented in the article by Amin Toosi et al. [29], we will also calculate the execution time of the proposed protocol. In the article by Amin Toosi et al. [29], the execution time of each operator is calculated and is given in Table 3. Also, in Table 4, the execution time of the proposed protocol in each section can be displayed.

Table 3: Execution time of various operators

Description	Symbol	Execution time (milliseconds)
Hash function execution time	T_h	0.0004
Symmetric encryption and decryption execution time	$T_{en/d}$	0.1303
Scalar multiplication execution time	T_{mu}	7.3529
Execution time of scalar addition	T_{ad}	0.1303
Biometric hash execution time	T_{bh}	0.01

Table 4: Execution time of the proposed protocol

	Authentication and Key Agreement Section	Storage section	Sharing section
Operators	$10T_h$	$10T_h + 2T_{en/d} + 4T_{mu}$	$8T_{en/d} + 4T_{mu}$
Runtime	0.004	29.6762	30.454

6. Conclusion and future work

In this research, our primary goal was to secure the stored information of entities participating in a supply chain and preserve their privacy. We presented a scheme that considers information security and privacy aspects in different parts, including information storage, access control, and information sharing. We can guarantee the confidentiality of participants' data and their privacy in the supply chain.

In line with the stated goal, we designed a protocol for each part, namely the access control and two-way authentication, storage, and information sharing. Then we showed in formal and informal ways that the proposed scheme meets various security needs and is resistant to known attacks. In our proposed scheme, we evaluated our proposed protocol using the formal and well-known Scyther tool, and the results obtained from this tool indicate that the proposed protocol is secure against various attacks and meets various security needs. Next, we measured the execution time of the proposed protocol by considering the execution time of each operator in the protocol, and it was found that the proposed scheme also has good efficiency in terms of execution time, and its practical implementation is also possible. As future work, it is possible to use the concept of blockchain in the supply chain more in future schemes, and security schemes such as searchable encryption can also be used to determine what information the requester has about the data owner.

References

[1] S. Chopra and P. Meindl, "Supply Chain Management. Strategy, Planning & Operation," in Das Summa Summarum des Management, C. Boersch and R. Elschen, Eds. Wiesbaden: Gabler, 2007, pp. 265–275.
[2] C. Chauhan and A. Singh, "A review of Industry 4.0 in supply chain management studies," J. Manuf. Technol. Manag., vol. 31, no. 5, pp. 863–886, Nov. 2019.
[3] G. F. Frederico, J. A. Garza-Reyes, A. Anosike, and V. Kumar, "Supply Chain 4.0: concepts, maturity and research agenda," Supply Chain Manag. An Int. J., vol. 25, no. 2, pp. 262–282, Sep. 2019.
[4] A. Shishodia, R. Sharma, R. Rajesh, and Z. H. Munim, "Supply chain resilience: A review, conceptual framework and future research," Int. J. Logist. Manag., vol. 34, no. 4, pp. 879–908, Jun. 2023.
[5] J. Sunny, N. Undralla, and V. Madhusudanan Pillai, "Supply chain transparency through blockchain-based traceability: An overview with demonstration," Comput. Ind. Eng., vol. 150, p. 106895, Dec. 2020.
[6] M. H. Abidi, H. Alkhalefah, U. Umer, and M. K. Mohammed, "Blockchain-based secure information sharing for supply chain management: Optimization assisted data sanitization process," Int. J. Intell. Syst., vol. 36, no. 1, pp. 260–290, Jan. 2021.
[7] S. Kumar, H. Banka, and B. Kaushik, "Ultra-lightweight blockchain-enabled RFID authentication protocol for supply chain in the domain of 5G mobile edge computing," Wirel. Networks, vol. 29, no. 5, pp. 2105–2126, Jul. 2023.
[8] M. Warren and W. Hutchinson, "Cyber attacks against supply chain management systems: a short note," Int. J. Phys. Distrib. Logist. Manag., vol. 30, no. 7/8, pp. 710–716, Sep. 2000.
[9] C. Zhang and S. Li, "Securing Information Sharing in Internet-Based Supply Chain Management Systems," Comput. Inf. Syst. Work. Pap., vol. 46, Jun. 2010.
[10] S. P N and A. S. Kunnathur, "Information security in supply chains: a management control perspective," Inf. Comput. Secur., vol. 23, no. 5, pp. 476–496, Nov. 2015.
[11] G. E. Smith, K. J. Watson, W. H. Baker, and J. A. Pokorski II, "A critical balance: collaboration and security in the IT-enabled supply chain," Int. J. Prod. Res., vol. 45, no. 11, pp. 2595–2613, Jun. 2007.
[12] D. Y. Zhang, Y. Zeng, L. Wang, H. Li, and Y. Geng, "Modeling and evaluating information leakage caused by inferences in supply chains," Comput. Ind., vol. 62, no. 3, pp. 351–363, Apr. 2011.
[13] K. Fan, Q. Luo, K. Zhang, and Y. Yang, "Cloud-based lightweight secure RFID mutual authentication protocol in IoT," Inf. Sci. (Ny)., vol. 527, pp. 329–340, Jul. 2020.
[14] J. H. Khor and M. Sidorov, "Security Flaws and Improvement of a Cloud-Based Authentication Protocol for RFID Supply Chain Systems," in 2018 3rd International Conference on Computer and Communication Systems (ICCCS), 2018, pp. 487–491.
[15] X. Li, "Inventory management and information sharing based on blockchain technology," Comput. Ind. Eng., vol. 179, p. 109196, May 2023.
[16] G. Zhang, Z. Yang, and W. Liu, "Blockchain-based decentralized supply chain system with secure information sharing," Comput. Ind. Eng., vol. 182, p. 109392, Aug. 2023.
[17] D. Preuveneers, W. Joosen, and E. Ilie-Zudor, "Trustworthy data-driven networked production for customer-centric plants," Ind. Manag. Data Syst., vol. 117, no. 10, pp. 2305–2324, Dec. 2017.
[18] Z. Gao, L. Xu, L. Chen, X. Zhao, Y. Lu, and W. Shi, "CoC: A Unified Distributed Ledger Based Supply Chain Management System," J. Comput. Sci. Technol., vol. 33, no. 2, pp. 237–248, Mar. 2018.
[19] M. Sidorov, M. T. Ong, R. V. Sridharan, J. Nakamura, R. Ohmura, and J. H. Khor, "Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains," IEEE Access, vol. 7, pp. 7273–7285, 2019.
[20] S. Jangirala, A. K. Das, and A. V Vasilakos, "Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment," IEEE Trans. Ind. Informatics, vol. 16, no. 11, pp. 7081–7093, Nov. 2020.
[21] V. Kumar and S. K. Das, "Enhancing security in IIoT: RFID authentication protocol for edge computing and blockchain-enabled supply chain," Cyber Secur. Appl., vol. 3, p. 100087, Dec. 2025.
[22] S. K. Dwivedi, R. Amin, and S. Vollala, "Blockchain based secured information sharing protocol in supply chain

management system with key distribution mechanism," J. Inf. Secur. Appl., vol. 54, p. 102554, Oct. 2020.

[23] L. Bader, J. Pennekamp, R. Matzutt, D. Hedderich, M. Kowalski, V. Lücken, and K. Wehrle, "Blockchain-based privacy preservation for supply chains supporting lightweight multi-hop information accountability," Inf. Process. Manag., vol. 58, no. 3, p. 102529, May 2021.

[24] C. Liu, F. Xiang, and Z. Sun, "Multiauthority Attribute-Based Access Control for Supply Chain Information Sharing in Blockchain," Secur. Commun. Networks, vol. 2022, no. 1, pp. 1–18, Apr. 2022.

[25] E. Gómez-Marín, V. Senni, L. Parrilla, J. L. Tejero López, E. Castillo, and D. Martintoni, "An Innovative Strategy Based on Secure Element for Cyber–Physical Authentication in Safety-Critical Manufacturing Supply Chain," Appl. Sci., vol. 13, no. 18, p. 10477, Sep. 2023.

[26] C. J. F. Cremers, "Scyther : semantics and verification of security protocols." Technische Universiteit Eindhoven, Eindhoven, 2006.

[27] M. Wazid, A. K. Das, N. Kumar, and M. Alazab, "Designing Authenticated Key Management Scheme in 6G-Enabled Network in a Box Deployed for Industrial Applications," IEEE Trans. Ind. Informatics, vol. 17, no. 10, pp. 7174–7184, Oct. 2021.

[28] D. Tiwari, G. K. Chaturvedi, and G. R. Gangadharan, "ACDAS: Authenticated controlled data access and sharing scheme for cloud storage," Int. J. Commun. Syst., vol. 32, no. 15, p. e4072, Oct. 2019.

[29] H. Amintoosi, M. Nikooghadam, M. Shojafar, S. Kumari, and M. Alazab, "Slight: A lightweight authentication scheme for smart healthcare services," Comput. Electr. Eng., vol. 99, p. 107803, Apr. 2022.