

A Lightweight Authentication Framework for Scalable Blockchain-Enabled IoT Supply Chain Systems

Umi Nabilah Badrol Said¹, Mohd Rizuan Baharon^{2*}, Hairol Nizam Mohd Shah³, Áine MacDermott⁴

Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia^{1,2}

Fakulti Teknologi & Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia³

School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, L3 3AF, United Kingdom⁴

Abstract—The rapid growth of IoT technologies and blockchain has increased attention to IoT–blockchain integration in various environments. A blockchain is a distributed ledger that records encrypted blocks and stores transactions or digital events shared among entities, which can be verified in the future. Blockchain integration is expected to improve supply chains through record immutability, enhanced traceability, quality assurance, decentralised control, and reduced fraud risks. However, scalability issues create challenges in terms of computational power and mining complexity due to the limited storage capacity of IoT devices, resulting in trust issues among stakeholders. This research proposes a mechanism to address sensor scalability and data privacy issues through a lightweight authentication algorithm for IoT sensors in supply chains. The study aims to develop an enhanced lightweight authentication algorithm to validate legitimate and fraudulent nodes and integrate it with blockchain technology. The proposed algorithm uses bitwise operations and hash functions to improve scalability, security, and data integrity.

Keywords—Authentication; blockchain; Internet of Things (IoT); lightweight security; supply chain; data privacy

I. INTRODUCTION

A supply chain is a network of distribution facilities and providers, including producers, dealers, and resellers, responsible for procuring raw materials, processing them into intermediate and finished products, and distributing finished products to consumers. The primary aim of a supply chain is to optimise the overall value generated. Supply chain management is therefore essential to ensure the efficient flow of materials and finished products from manufacturers to consumers, utilising manufacturing facilities and warehouses throughout the process. Manufacturers, suppliers, transporters, warehouses, retailers, and customers represent all stages involved in the manufacturing industry, either directly or indirectly, in fulfilling customer demand [1]. In recent years, supply chain analysis has gained importance as it provides insights into labour division practices, increasing competition, global market entry that supports sustainable income growth, and the realisation of globalisation benefits [1]–[3]. As supply chains play a significant role in the global fish industry, transparency is a fundamental requirement.

Although the Internet of Things (IoT) offers substantial benefits and addresses a wide range of challenges across various industries, several difficulties remain. These challenges mainly involve security threats, privacy concerns, and system vulnerabilities. By analysing IoT component interactions, this section briefly discusses the potential issues associated with IoT systems. Most IoT-related challenges are closely linked to security and privacy concerns, and security remains a critical requirement for these applications. Despite the rapid growth of IoT technologies, security issues have not been comprehensively addressed. Both industry professionals and academics regard security as the most critical challenge due to inherent IoT vulnerabilities. Many existing security solutions are unsuitable for IoT systems because they impose high energy consumption and computational overhead [4], [5]. Furthermore, threats such as data theft, physical tampering, hacking, and counterfeiting among supply chain partners contribute to trust issues. According to [6], IoT systems must be protected against external attacks, support secure data aggregation at the perception and network layers, and ensure that only authorised entities can access or modify information at the application layer.

IoT-enabled smart agriculture systems are particularly vulnerable to a variety of attacks [7], [8]. A replay attack occurs when an attacker intercepts a secure communication session, gains access to transmitted data, and fraudulently delays or resends it to deceive a legitimate entity. A man-in-the-middle attack takes place during communication between two users, allowing an attacker to intercept or alter messages without detection. A stolen-verifier attack arises when an attacker extracts authentication information from a gateway node that stores identity-related credentials. In a stolen or lost smart card attack, attackers obtain a smart card or mobile device and extract credentials using power analysis or timing attacks. The retrieved credentials may then be exploited to access sensitive data. Password guessing attacks involve attempting to derive correct credentials from intercepted messages, while password change attacks require access to compromised devices to alter legitimate users' passwords. A denial-of-service (DoS) attack disrupts services by exhausting system resources, causing hardware or software malfunctions. Privileged insider attacks occur when authorised users misuse their access rights to gain unauthorised

*Corresponding author.

modification or deletion privileges. Identity theft attacks involve adversaries impersonating legitimate users, making it difficult to distinguish between genuine and fraudulent messages.

Resilience against sensing device capture attacks is another critical concern. Such attacks occur when an IoT sensing device is physically captured, enabling attackers to extract confidential data and disrupt network communication. The impact of device capture attacks can be estimated by calculating the probability of compromised communication. These threats must be considered in system authentication, key management, and access control mechanisms. The objective is to minimise the probability of successful attacks, ideally approaching zero. Other IoT vulnerabilities include wormhole attacks, Sybil attacks, unauthorised deployment of sensing devices, and node replication attacks. Wormhole attacks use in-band or out-of-band channels to connect distant nodes, misleading them into believing that they are in proximity and granting attackers control over network traffic. Sybil attacks involve the creation of multiple fake identities, enabling attackers to submit multiple requests under different identities. Similarly, node replication attacks involve duplicating compromised devices to infiltrate the network and interact with legitimate nodes.

The emergence of blockchain technology in supply chain applications has attracted considerable attention due to its decentralised and tamper-resistant properties [9], [10]. However, alongside technological advancements, issues such as data tampering, fraud, falsification, and trust remain significant concerns before and after data are recorded on the blockchain [11]–[14]. Authentication and validation of data sources continue to be challenging tasks [15]. Consequently, identifying and authenticating fraudulent nodes within blockchain-based infrastructures remain difficult, highlighting the need for robust security mechanisms. To address these challenges, an enhanced lightweight authentication algorithm is proposed to support IoT-enabled supply chains while preserving data privacy.

Despite extensive prior research, several issues remain unresolved. Studies focusing on scalability often neglect security aspects, while security-focused approaches frequently compromise scalability. Most existing research does not adequately address system scalability, even though supply chains involve large and complex networks comprising numerous entities. As network size increases, communication overhead grows, leading to higher energy consumption and reduced device lifespan due to the limited computational capabilities of IoT devices. Addressing these challenges can benefit industries by enabling secure and efficient supply chain operations that support economic growth and product safety. The proposed framework aims to identify and eliminate unauthorised or compromised nodes while ensuring that only legitimate nodes participate in the supply chain. Accordingly, the objectives of this study are to identify key features for authentication and validation, design a lightweight authentication algorithm to distinguish legitimate and fraudulent nodes, and evaluate the performance of the proposed algorithm through a proof-of-concept implementation.

Recent systematic reviews and empirical studies confirm that integrating blockchain technology with IoT systems enhances security, trust, and identity management, particularly

in large-scale decentralised environments [10]. Blockchain-based frameworks support decentralised identity management, smart contract-driven access control, and tamper-proof audit trails, addressing many security challenges in heterogeneous IoT systems. Nevertheless, scalability remains a major obstacle in densely populated IoT networks. Lightweight consensus protocols such as Delegated Proof of Stake (DPoS) have demonstrated improved throughput and reduced latency in large-scale deployments [16]. Similarly, lightweight authentication frameworks for resource-constrained domains, including the Internet of Medical Things, have been proposed using hybrid blockchain and edge computing architectures to balance security and scalability [17]. Recent studies integrating lattice-based cryptography with blockchain further demonstrate reduced computational overhead and improved resistance to cryptanalytic attacks [18]. Despite these advances, few solutions provide a holistic integration of lightweight authentication and scalable blockchain mechanisms for supply chain environments.

The novelty of this study lies in the joint optimisation of lightweight authentication and scalable blockchain integration for IoT-enabled supply chains. By employing simplified bitwise operations and efficient hash functions, the proposed approach minimises computation and storage overhead at the sensor level while ensuring that authenticated transactions are securely and immutably recorded on the blockchain ledger. The objectives of this research are to identify authentication features, design an efficient lightweight authentication algorithm, and evaluate its performance through a proof-of-concept implementation. The significance of this study lies in its contribution to a cost-effective and scalable solution that enhances trust, data integrity, and the secure participation of heterogeneous IoT devices, thereby supporting the broader adoption of blockchain technologies in supply chain systems.

The remainder of the study is structured as follows: Section II outlines the methodology of the proposed scheme. Section III presents the results and discussion. Finally, the conclusion is provided in Section IV.

II. THE PROPOSED AUTHENTICATION ALGORITHM

This section presents the design of a lightweight authentication algorithm for an IoT sensor node. There are two primary components of the authentication algorithm:

TABLE I. THE NOTATIONS USED IN THE AUTHENTICATION ALGORITHM

Symbols	Description
T, R, S	Box sensor, reader/gateway, and supply chain nodes, respectively
ID _T , ID _R , ID _S	Identity of the box sensor, the reader/gateway, and the supply chain node, respectively
K _{RT} , K _{RS}	A shared key between T and R, R and S for identity validation, respectively
h(.)	SHA-256
R _N , R _R , R _S	Random nonces
T _K , T _T , T _R , T _S	Current timestamps
ΔT	Maximum transmission delay
Rot(X, Y)	Left rotation of X by the Hamming weight of Y
RRot(X, Y)	X $\gg$ Y, right rotation of X by the Hamming weight of Y

The initialization phase and the authentication and key establishment phase. Table I presents the notations used in the authentication algorithm, along with their definitions.

A. Initialisation Phase

The initialisation phase begins when the raw material arrives at the distribution centre, as shown in Fig. 1. This research considers the box sensor, reader/gateway, and supply chain server identities as secret keys. Once a box sensor is registered in the supply chain system, it permanently stores the box sensor ID (ID_T) and the supply chain server ID (ID). Subsequently, the reader/gateway ID (ID_R) is sent to the box sensor (T) via $ID_C = ROT(ID_T \oplus ID_R \oplus T_K, ID_T \oplus T_K)$, where T_K represents the current timestamp of the operation. The box sensor then stores the tuple $\{ID_T, ID_R, IDs\}$. Through a secure channel, the reader/gateway (R) receives the supply chain system ID (IDs) along with a list of box sensor IDs (ID_T) assigned to the specific reader/gateway (R) from the supply chain system. Accordingly, the reader/gateway (R) stores the tuple $\{ID_T, ID_R, IDs\}$. The reader/gateway (R) is assumed to be a high-end device with greater computational capability compared to the box sensor (T). All devices used in this research are assumed to be secured and equipped with a Trusted Platform Module (TPM).

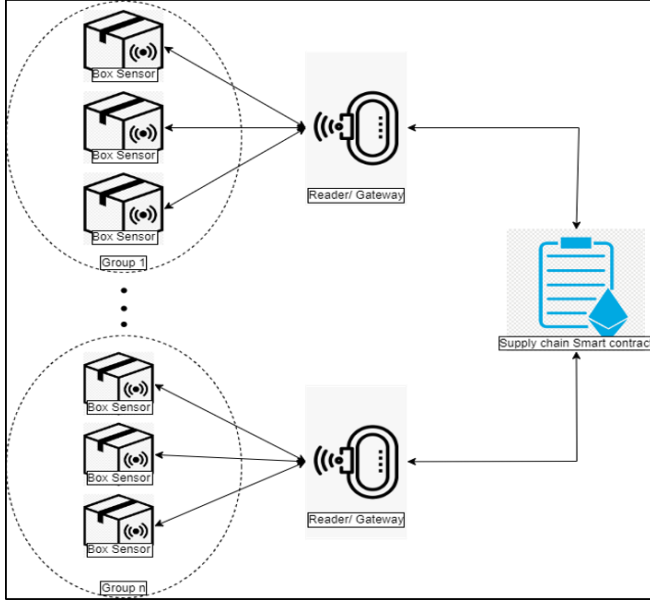


Fig. 1. The placement of the device in the supply chain.

B. Implementation of the Proposed Authentication Algorithm

The proposed authentication algorithm is implemented in the simulation environment to evaluate its security strength and to calculate its power consumption. The proposed algorithm consists of Tag (T) Initialization, Reader (R) Verification, Supply Chain Node (S) Verification, Reader (R) Final Authentication, and Tag (T) Final Computation. The overall steps involved in the proposed work are explained below (Algorithms 1-5).

Algorithm 1: Tag (T) Initialization

```
 $R_T \leftarrow RAND()$   
 $T_T \leftarrow NOW()$   
 $KR_T \leftarrow h(ID_T || ID_R)$ 
```

```
 $M_T \leftarrow ROT((R_T \oplus ID_T) \oplus T_T), (ID_R \oplus ID_T)$   
 $C_T \leftarrow h(M_T || |ID_T| || R_T)$   
 $MSG_1 \leftarrow \{KR_T, M_T, C_T, T_T\}$   
send( $MSG_1$ )
```

Algorithm 2: Reader (R) Verification

```
 $MSG_1 \leftarrow recv(R)$  #  $\{KR_T, M_T, C_T, T_T\}$   
if  $|NOW() - T_T| \geq \Delta T$  then  
    terminate("TT stale or replay detected")  
else, compute  $K'_{RT} \leftarrow h(ID_T || ID_R)$   
if  $K'_{RT} \neq KR_T$  then  
    terminate("Tag->Reader  $K_{RT}$  mismatch")  
else, ("retrieve  $ID_T$  and extract  $R_T$  from  $M_T$ ")  
     $R_T \leftarrow RROT(M_T, (ID_R \oplus ID_T)) \oplus T_T \oplus ID_T$   
Compute  $C'_T \leftarrow h(M_T || ID_T || T_T)$   
if  $C'_T \neq C_T$  then  
    terminate("Tag->Reader  $C_T$  mismatch")  
else, generate random nonce  $R_R$  and timestamp  $T_R$   
 $R_R \leftarrow RAND()$   
 $T_R \leftarrow NOW()$   
Retrieve  $K_{RS}$ ,  
     $K_{RS} \leftarrow h(ID_T || IDs || ID_R)$   
Compute  $M_R$ ,  
     $M_R \leftarrow ROT((R_R \oplus ID_R) \oplus T_R, IDs \oplus ID_R)$   
Retrieve  $C_R$ ,  
     $C_R \leftarrow h(M_R || ID_R || T_R)$   
 $MSG_2 \leftarrow \{K_{RS}, M_R, C_R, T_R\}$   
send( $MSG_2$ )
```

Algorithm 3: Supply Chain Node (S) Verification

```
 $MSG_2 \leftarrow recv(S)$  #  $\{K_{RS}, M_R, C_R, T_R\}$   
if  $|NOW() - T_R| \geq \Delta T$  then  
    terminate("TR stale or replay detected")  
retrieve  $K'_{RS}$ ,  
     $K'_{RS} \leftarrow h(ID_T || ID_R || IDs)$   
if  $K'_{RS} \neq K_{RS}$  then  
    terminate("Reader->Node  $K_{RS}$  mismatch")  
else, compute  $R_R$   
     $R_R \leftarrow RROT(M_R, (IDs \oplus ID_R)) \oplus T_R \oplus ID_R$   
Retrieve  $C'_R$ ,  
     $C'_R \leftarrow h(M_R || ID_R || T_R)$   
if  $C'_R \neq C_R$  then  
    terminate("Reader->Node  $C_R$  mismatch")  
else, generate  $R_s$  at  $T_s$   
 $R_s \leftarrow RAND()$   
 $T_s \leftarrow NOW()$   
Compute  $M_s$ ,  
     $M_s \leftarrow ROT(R_s, IDs) \oplus ROT(ID_s, ID_R \oplus T_s)$   
Compute session key  $SK_{ST}$ ,  
     $SK_{ST} = h(ID_T \oplus IDs \oplus R_s \oplus T_s)$   
Retrieve  $S_R$ ,  
     $S_R \leftarrow h(R_s || ID_R || IDs || T_s)$   
Retrieve  $S_T$ ,  
     $S_T \leftarrow h(R_s || ID_T || ID_R || IDs || T_s)$   
 $MSG_3 \leftarrow \{M_s, S_R, S_T, T_s\}$   
send( $MSG_3$ )
```

Algorithm 4: Reader (R) Final Authentication

```
 $MSG_3 \leftarrow recv(R)$  #  $\{M_s, S_R, S_T, T_s\}$   
if  $|NOW() - T_s| \geq \Delta T$  then  
    terminate("Ts stale or replay detected")  
else, compute  $R_s$ 
```

```
RS ← RROT((MS ⊕ ROT(IDS, IDR ⊕ TS)), IDR)
Retrieve S'R,
    S'R ← h(RS || IDR || IDS || TS)
if S'R ≠ SR then
    terminate("Node->Reader SR mismatch")
compute MP,
    MP ← ROT(RS, IDR) ⊕ ROT(RT, TS)
MSG4 ← {MP, ST, TS}
send(MSG4)
```

Algorithm 5: Tag (T) Final Computation

```
MSG4 ← recv(T) # {MP, ST, TS}
Compute RS,
    RS ← RROT((MP ⊕ ROT(RT, TS)), IDR)
Compute S'T,
    S'T ← h(RS || IDT || IDR || IDS || TS)
if S'T ≠ ST then
    terminate("Reader->Tag ST mismatch")
Derive the session key SKST,
    SKST ← h(IDT ⊕ IDS ⊕ RS ⊕ TS)
stores the session key (S),
    store(SKST)
uses the session key (S),
    use(SKST)
```

III. RESULT AND DISCUSSION

Section III discusses the threat model, the security analysis, and the performance analysis of the suggested method.

A. Threat Model

Two types of threat models were chosen: the 'Dolev-Yao threat model' and 'Canetti and Krawczyk's adversary model'. This research employed the well-known "Dolev-Yao (DY) threat model," in which an adversary can perform several potential attacks, including impersonation, replay, man-in-the-middle, and ephemeral secret leakage (ESL) attacks, as well as eavesdrop, modify, delete, or incorporate fake information between the various entities' communications [19]. The CK-adversary model enables the adversary to compromise the session states together with secret information, including secret keys, in addition to his/her abilities as accomplished utilising the DY model. Considering this, regardless of whether session states and secret information are revealed in a single session, the compromised information must not jeopardise the privacy of other parties' login information. The forward and backward secrecy of an authentication scheme must therefore be ensured under the CK-adversary model [20]. This section demonstrates how the suggested approach resists various threats and guarantees user anonymity and confidentiality. The analysis was performed using AVISPA for formal security verification, and theoretical evaluations were conducted as informal security verification [21], [22]:

B. Informal Security Analysis

Comparing the attack resistance and security features of similar algorithms is a crucial step in identifying the most robust and resilient solution for IoT-blockchain supply chain environments [23], [24]. This comparative analysis examines how well each algorithm withstands attacks such as brute force, cryptographic, or network-based attacks. Additionally, it evaluates resistance to known vulnerabilities and potential

weaknesses, highlighting the ability to maintain data integrity and confidentiality. Moreover, the comparison reviews security features provided by each algorithm, including authentication methods, key management, and data protection mechanisms, as provided below:

1) *Confidentiality*: Messages exchanged $T \rightarrow R$: $MSG_1 = \{K_{RT}, M_T, C_T, T_T\}$, $MSG_2 = \{K_{RS}, M_R, C_R, T_R\}$, $MSG_3 = \{M_S, S_R, S_T, T_S\}$, $MSG_4 = \{M_P, S_T, T_S\}$ are correlated with participant-generated random numbers (R_T , R_R , and R_S). It is computationally infeasible for an attacker (A) to extract R_T , R_R , and R_S to compute valid messages and impersonate legitimate participants. Therefore, transmitted data confidentiality is preserved.

2) *User anonymity*: If attacker A can intercept the messages MSG_i , where $i = 1, 2, 3$, and 4, that were transmitted across unsecured channels when the process of authentication and key exchange takes place. It is computationally impossible for A to predict the identities of the box sensor (T) and reader (R) in polynomial time without knowing the parametric values ID_S , ID_T , and R_N . This guarantees that the user anonymity attribute of the suggested method is maintained.

3) *Mutual authentication and session key establishment*: The proposed method involves mutual authentication between the reader (R), the box sensor (T), and the supply chain node (S). The secret random number R_N and the present timestamp T_T are used to protect the message $MSG_1 = \{K_{RT}, M_T, C_T, T_T\}$. By checking that $C'_T = h(M_T || ID_T || R_T)$, only the genuine R can acknowledge. R then instructs S to transmit the message $MSG_2 = \{K_{RS}, M_R, C_R, T_R\}$. S verifies R after receiving the message using $C_R = h(R_R || ID_R || T_R)$. The message $MSG_3 = \{M_S, S_R, S_T, T_S\}$ is then transmitted by S after it creates its random number R_S and the present timestamp T_S and computes the session key $SK_{ST} = h(ID_T \oplus ID_S \oplus R_S \oplus T_S)$. When R receives a message from S , $S'_R = h(R_S || ID_R || T_S)$ is used to validate the message and confirm S'_R validity. When the verification is successful, R authenticates S and sends the information $MSG_4 = \{M_P, S_T, T_S\}$ to T . Upon receiving the MSG_4 , T derives the random number R_S from MP . T then uses R_S to check for the authenticity of R and S by performing $S'T = h(R_S || ID_T || ID_R || T_S)$. In addition, all participants are successful in confirming the legitimacy of the sender and receiver parties. If the verification is successful, T computes the session key SK_{ST} and enables T and S to establish a secure connection. As a result, the suggested approach is effective in maintaining mutual authentication and guaranteeing the generation of the session key.

4) *Forward secrecy*: If an attacker A compromises the box sensor T , the secret keys held within it are likewise revealed. The data sent during the prior session may then be obtained by A . The reader ID and the random numbers are not saved in the tag's memory in the suggested method. Additionally, to ensure that the session keys are unique for each session, new random numbers are created for each session. Therefore, A would not benefit in any way from the session key or random numbers in a particular session being compromised. As a result, the offered approach guarantees forward secrecy.

5) *Internal attacks*: Internal participants may attempt impersonation. Two cases are considered:

a) *Tag impersonation attack*: We also take into account the possibility that an internal, legitimate box sensor from store 1 (S_1) that has access to secret parameters tries to mimic a box sensor from a different S_2 that has secret parameters distinct from S_1 's. The fraudulent box sensor sends MSG_1 to the genuine reader, which belongs to S_2 , when it inquires about the reader. Because the ID_T saved in S_2 differs from the box sensor ID available to reader S_1 , the message MSG_1 cannot be validated by reader S_1 . This indicates that the reader is unable to determine the validity of MSG_1 by examining the condition $C'_T = h(M_T \parallel ID_T \parallel R_N)$, as well as extract the proper R_T . The tag impersonation attack is constrained as a result.

b) *Reader impersonation attack*: We assume that a legitimate internal reader of store 1 (S_1) attempts to mimic the reader of store 2 (S_2) with its own secret settings. Now, the fraudulent reader may obtain MSG_1 from the genuine T , which belongs to S_2 , when it queries the box sensor. However, the reader S_1 was unable to retrieve the appropriate ID_T in order to extract the random number produced by T ; hence, the message MSG_1 could not be authenticated. Since the reader is unable to retrieve the right ID_T and R_N , and the ID_T value is invalid, it is unable to continue the supply chain node authentication process. Thus, the suggested method is to prevent reader impersonation attacks.

6) *External attacks*: In this research, we took into consideration two types of external attacks.

a) *Replay attack*: It is assumed that attacker A attempts to intercept messages $MSG_i, i \in [1, 5]$, and replay them to the recipient during the authentication and key exchange phase. However, this attempt fails because the transmitted messages include current timestamps and random numbers. Timestamp verification occurs first after the messages are received, followed by validation of the transmitted messages. Thus, the proposed method effectively resists replay attacks.

b) *Man-in-the-middle attack*: Assume that attacker A tries to record and alter messages $MSG_i, i \in [1, 5]$, during the authentication and key exchange phase so that participants believe the messages originate from legitimate entities. For example, if A attempts to modify MSG_1 to launch this attack, the attempt fails due to the lack of knowledge of the secret R_T . Similarly, subsequent messages— MSG_2 (random secret R_R), MSG_3 (random secret R_S), and MSG_4 (random secrets R_S and R_T)—cannot be modified despite A 's efforts. Therefore, the proposed method withstands man-in-the-middle attacks because all messages incorporate random values and current timestamps.

7) *Ephemeral Secret Leakage (ESL) attack*: After mutual authentication is verified, the supply chain node (S) and the tag (T) establish a shared session key SK_{ST} during the authentication and key exchange phase. The current CK-adversary model for session key security is considered in accordance with the threat model discussed in Section I. Two cases illustrate the robustness of the proposed method:

a) *Case 1*: Assume that attacker A knows the ephemeral (short-term) secrets R_N, R_R , and R_S . A 's challenge is to compute SK_{ST} based on these short-term secrets. However, this is infeasible because A cannot guess the long-term secrets (ID_T, ID_S , and ID_R), which is computationally infeasible.

b) *Case 2*: Consider that A obtains some or all long-term secrets (ID_T, ID_S , and ID_R). Building SK_{ST} remains A 's challenge, similar to Case 1. However, predicting the correct short-term secrets (R_N, R_R , and R_S) without knowing their actual values is also computationally infeasible.

From these cases, it is evident that SK_{ST} can only be computed using both valid long-term and short-term secrets, which is possible only for legitimate participants (S, R , and T). Furthermore, because the proposed method uses random and distinct session keys for each session, compromising the current session key does not affect previous or future sessions. Even if the current session key is exposed, A cannot derive past or future keys. Thus, the proposed approach successfully maintains SK -security and ensures both backward and forward secrecy. At most, one session key could be compromised through session hijacking, but its impact does not jeopardize other sessions. Therefore, the proposed method is protected against ESL attacks.

C. Formal Security Analysis

To provide a comprehensive and unbiased formal examination of the proposed method, the security of the protocol was analysed using a combination of AVISPA and Scyther tools. This combined approach mitigates attack omissions caused by limitations in security models and algebraic property handling within AVISPA and Scyther, respectively.

1) *Analysis of AVISPA results*: The intruder simulation under OFMC is shown in Fig. 2, while the analysis results are illustrated in Fig. 3 and 4. The findings indicate that AVISPA's OFMC engine and CL-AtSe engine report the protocol as 'safe' within 0.02 seconds and a similarly short duration, respectively. In addition, the extensive intruder knowledge set indicates that an attacker can learn multiple hashed nonce structures; therefore, the formal verification step (e.g., using AVISPA backends) is essential to confirm that, despite this knowledge, the authentication and secrecy goals remain valid, and no feasible attack trace exists.

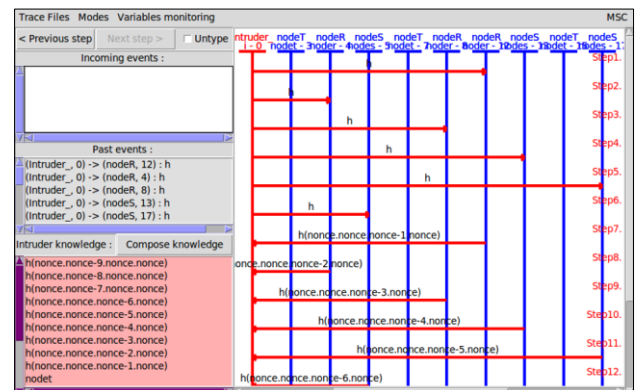


Fig. 2. The intruder simulation under OFMC.



Fig. 3. Result of the simulation under OFMC.

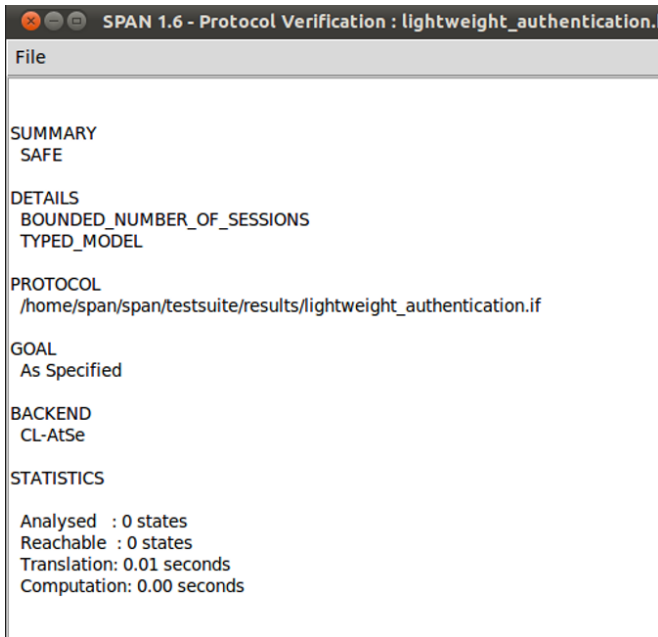


Fig. 4. Result of the simulation under OFMC and CL-AtSe backend.

D. Analysis of Scyther Results

The Scyther-Compromise tool is used to analyse the authentication algorithm under the CK security framework by examining the following: "Session-Key Reveal," "Long-Term Key Reveal," "State Reveal," and "wPFS" in the statistical options, as shown in Fig. 5. The tool shows that, under the CK model, the protocol is session-key secured.

1) *Performance analysis*: Conducting a comprehensive performance analysis of the enhanced lightweight authentication algorithm and comparing it with other lightweight authentication algorithms is essential for evaluating its effectiveness in the IoT-blockchain supply chain environment. This analysis involves assessing key metrics, including communication cost, computational cost, and storage cost. By systematically measuring and comparing these factors, researchers and practitioners gain valuable insights into the algorithm's efficiency and suitability for real-world

deployment. Such evaluation enables a precise assessment of how well the enhanced algorithm balances security requirements and resource constraints inherent in IoT devices compared to other available options. Ultimately, this analysis not only validates the algorithm's performance but also supports informed decisions regarding its adoption and optimization, ensuring compliance with the stringent requirements of the evolving IoT-blockchain supply chain ecosystem.

Scyther results : verify				Status	Comments
Claim	T				
LightweightAuthentication	T	LightweightAuthentication_T1	Commit S_h(con(con(IDCRS3_ID3TS))	OK	No attacks within bounds.
		LightweightAuthentication_T2	Alive	OK	No attacks within bounds.
		LightweightAuthentication_T3	Weakagree	OK	No attacks within bounds.
		LightweightAuthentication_T4	Niagree	OK	No attacks within bounds.
		LightweightAuthentication_T5	Niynch	OK	No attacks within bounds.
		LightweightAuthentication_T6	Secret IDr	OK	No attacks within bounds.
		LightweightAuthentication_T7	Secret IDs	OK	No attacks within bounds.
		LightweightAuthentication_T8	Secret IDt	OK	No attacks within bounds.
		LightweightAuthentication_T9	Secret SKst	OK	No attacks within bounds.
		LightweightAuthentication_T10	Secret RT	OK	No attacks within bounds.
		LightweightAuthentication_T11	Secret RS'	OK	No attacks within bounds.
	R	LightweightAuthentication_R1	Commit T_h(con(con(MUID1_RT'))	OK	No attacks within bounds.
		LightweightAuthentication_R2	Commit S_h(con(con(IDLRS1_IDr1TS))	OK	No attacks within bounds.
		LightweightAuthentication_R3	Alive	OK	No attacks within bounds.
		LightweightAuthentication_R4	Weakagree	OK	No attacks within bounds.
		LightweightAuthentication_R5	Niagree	OK	No attacks within bounds.
		LightweightAuthentication_R6	Niynch	OK	No attacks within bounds.
		LightweightAuthentication_R7	Secret IDr	OK	No attacks within bounds.
		LightweightAuthentication_R8	Secret IDs	OK	No attacks within bounds.
		LightweightAuthentication_R9	Secret IDt	OK	No attacks within bounds.
		LightweightAuthentication_R10	Secret RR	OK	No attacks within bounds.
		LightweightAuthentication_R11	Secret RT'	OK	No attacks within bounds.
		LightweightAuthentication_R12	Secret RS'	OK	No attacks within bounds.
	S	LightweightAuthentication_S1	Commit R_h(con(con(IDCRS3_TR))	OK	No attacks within bounds.
		LightweightAuthentication_S2	Weakagree	OK	No attacks within bounds.
		LightweightAuthentication_S3	Niagree	OK	No attacks within bounds.
		LightweightAuthentication_S4	Niynch	OK	No attacks within bounds.
		LightweightAuthentication_S5	Secret IDr	OK	No attacks within bounds.
		LightweightAuthentication_S6	Secret IDs	OK	No attacks within bounds.
		LightweightAuthentication_S7	Secret IDt	OK	No attacks within bounds.
		LightweightAuthentication_S8	Secret RR'	OK	No attacks within bounds.
		LightweightAuthentication_S9	Secret SKst	OK	No attacks within bounds.
		LightweightAuthentication_S10	Secret RS	OK	No attacks within bounds.

Fig. 5. Result of security protocol verification in the Scyther tool.

2) *Communication cost*: To calculate communication cost, we assume a 256-bit hash output (SHA-256), a 160-bit Hamming weight, 160-bit identities and random numbers, and a 32-bit timestamp. According to Sidorov and Mujahid's protocols, the "Hello" message is 160 bits long. Similarly, the "Query" message in Fan et al.'s LRMAPC and ULRMAPC protocols is assumed to be 160 bits. Under these assumptions, messages MSG_1 , MSG_2 , MSG_3 and MSG_4 require $(256 + 160 + 256 + 32) = 704$ bits, $(256 + 160 + 256 + 32) = 704$ bits, $(256 +$

$160 + 256 + 32 = 704$ bits, and $(256 + 160 + 32) = 448$ bits, respectively. Therefore, the total communication cost for the proposed method is 2560 bits for four message exchanges.

As shown in Fig. 6, the proposed method incurs a higher communication cost than the other schemes. This is because LBRAPS and Sidorov et al. employ smaller key sizes of 160 bits and 96 bits, respectively, which may increase the risk of brute-force attacks during authentication message exchanges. LBRAPS requires a total communication cost of 2240 bits because it uses SHA-1, which produces a 160-bit hash output. However, SHA-1 is no longer considered secure due to known cryptographic weaknesses, thereby reducing the overall security of the authentication scheme. If LBRAPS were implemented using SHA-256, its total communication cost would increase to 3008 bits. In that case, MSG_1 , MSG_2 , MSG_3 , MSG_4 and MSG_5 would require 448 bits, 704 bits, 704 bits, 704 bits, and 448 bits, respectively.

Furthermore, the proposed method outperforms other existing protocols in terms of security. Unlike Mujahid's scheme and the LRMAPC and ULRMAPC protocols, which were not designed for blockchain integration and did not establish a session key between entities, the proposed method and protocols such as LBRAPS and Sidorov were specifically created for blockchain-based environments.

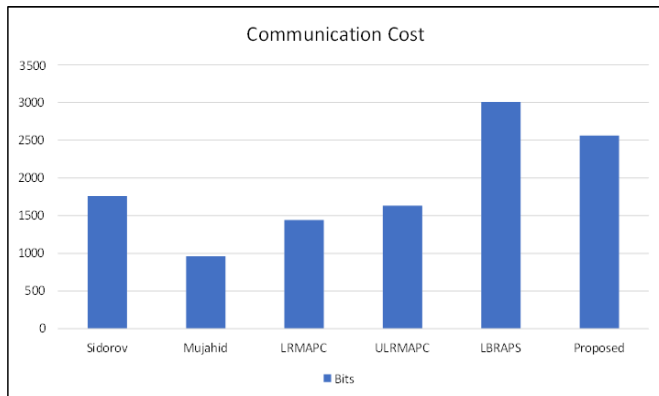


Fig. 6. Comparison of the communication cost between multiple algorithms.

3) *Computational cost*: The execution times for exclusive-OR, Hamming weight, one-way hash function, and left/right rotation operations—denoted as Txor, THW, THash, and TROT—are considered here. TROT and Txor are excluded from the computation because their impact is negligible. During the authentication and key exchange phase, a box sensor incurs a computational cost of $4THash + 2TROT + 2Txor$, a reader incurs $5THash + 4TROT + 4Txor$, and a supply chain node incurs $5THash + 2TROT + 2Txor$ during authentication and session key storage. Overall, the total computational cost is $14THash + 8TROT + 8Txor$. Based on the number of exchange messages in Fig. 7 and the comparative study of computational costs in Fig. 8, the proposed method requires less computational cost than the schemes by Jangirala et al. [25] and Sidorov et al. [26].

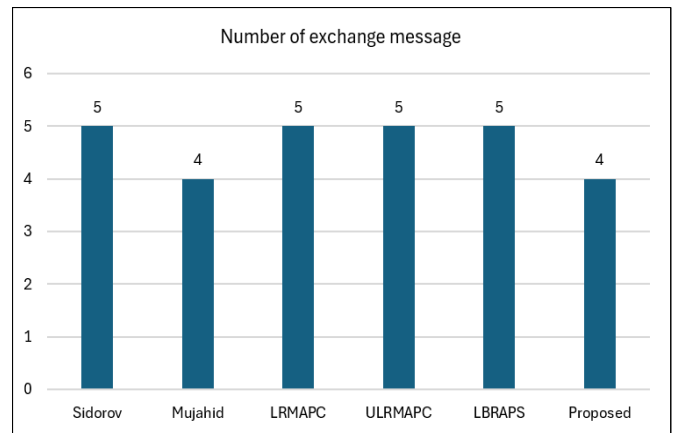


Fig. 7. Number of exchange messages.

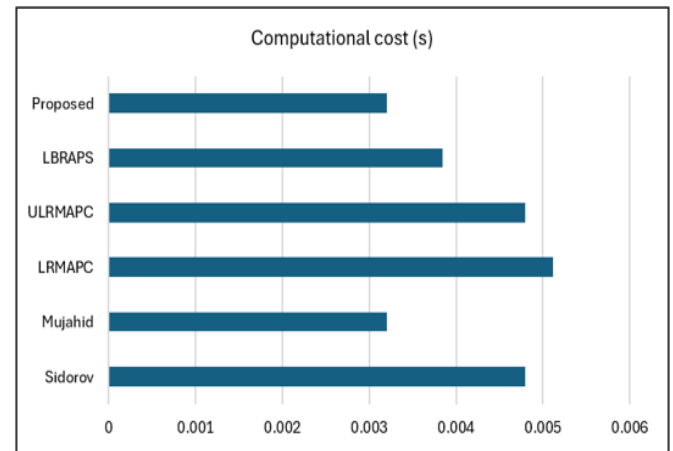


Fig. 8. Comparison of the computational cost between multiple algorithms.

IV. CONCLUSION

A lightweight authentication framework was proposed to address the critical challenges of integrating blockchain with IoT-enabled supply chains, where scalability, security, and resource limitations have traditionally posed significant barriers. Computation and storage overhead on resource-constrained IoT devices were minimised using simplified bitwise operations and efficient hash functions, while robust security features such as mutual authentication, user anonymity, and data confidentiality were ensured. Formal security verification was conducted using AVISPA and Scyther, and resilience against replay, man-in-the-middle, and ephemeral secret leakage attacks was confirmed. Informal analysis validated forward secrecy and protection against internal impersonation attempts. Performance evaluation indicated that the proposed method achieved a lower computational cost than existing schemes such as LBRAPS, Jangirala et al., and Sidorov et al., with only a marginal increase in communication cost due to the adoption of SHA-256 for enhanced security—a trade-off that significantly strengthened resistance to brute-force and cryptographic attacks. Secure session key establishment was supported, enabling end-to-end encrypted communication across supply chain nodes, a feature absent from several existing protocols.

Blockchain integration was employed to guarantee immutability and transparency, while the lightweight design ensured scalability for large IoT deployments, overcoming the limitations of prior approaches that compromised either security or performance. This dual optimisation provided a cost-effective, secure, and scalable architecture suitable for real-world supply chain environments, fostering trust, data integrity, and operational efficiency in decentralised ecosystems. Future enhancements will include the incorporation of post-quantum cryptographic primitives and adaptive consensus mechanisms to further improve resilience and performance. Overall, a practical and innovative solution was delivered that strengthens IoT–blockchain integration and paves the way for secure, transparent, and efficient supply chain operations in an increasingly connected world.

ACKNOWLEDGMENT

This work has been supported by the Universiti Teknikal Malaysia Melaka research grant, with the grant number ANTARABANGSA(IRMG)-AJMAN/2025/FTMK/A00110. The authors would like to express their gratitude to the Centre for Research and Innovation Management (CRIM) UTeM for its continuous support of this work.

REFERENCES

- [1] M. S. Islam, M. A. Khan, and M. A. Rahman, "Supply chain management in manufacturing industries," *Journal of Industrial Engineering*, vol. 6, no. 2, pp. 45–56, 2018.
- [2] G. Pedroza, J. L. Martinez, and R. Gomez, "Global supply chain analysis and its impact on sustainable income growth," *International Journal of Production Economics*, vol. 227, pp. 107–118, 2020.
- [3] M. Khayer, S. Akter, and A. Rahman, "Supply chain transparency and globalization benefits," *Journal of Business Logistics*, vol. 44, no. 1, pp. 23–39, 2023.
- [4] A. Rejeb, K. Rejeb, and H. G. Keogh, "Blockchain and IoT integration in supply chains: A review," *IEEE Access*, vol. 7, pp. 176–190, 2019.
- [5] A. Rejeb, K. Rejeb, and H. G. Keogh, "Advances in IoT security for supply chain applications," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2456–2468, 2023.
- [6] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture: Applications, challenges, and security," *Sensors*, vol. 20, no. 8, pp. 1–23, 2020.
- [7] A. Vangala, S. Roy, and A. K. Das, "Blockchain-based lightweight authentication protocol for IoT-enabled smart agriculture," 2022 International Conference on Cyber-Physical Social Intelligence (ICCSI), pp. 110–115, 2022.
- [8] S. Vangala, A. Das, and A. K. Jangirala, "Security challenges and solutions for IoT-based smart farming systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 4, pp. 1895–1908, 2022.
- [9] G. Probst, "Blockchain in supply chain management: Opportunities and challenges," *Supply Chain Management Review*, vol. 23, no. 4, pp. 12–19, 2019.
- [10] M. Khayer, S. Akter, and A. Rahman, "Blockchain integration in IoT-enabled supply chains: A systematic review," *Journal of Business Logistics*, vol. 45, no. 1, pp. 34–52, 2024.
- [11] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proc. 13th Int. Conf. Service Systems and Service Management (ICSSSM)*, 2017, pp. 1–6.
- [12] N. Kshetri, "Blockchain's roles in meeting key supply chain management objectives," *Int. J. Inf. Manage.*, vol. 39, pp. 80–89, 2018.
- [13] F. Pfreundt, "Blockchain for secure supply chain management," *Blockchain Research Journal*, vol. 2, no. 1, pp. 45–58, 2018.
- [14] W. Powell, M. Foth, S. Cao, and V. Natanelov, "Garbage in garbage out: The precarious link between IoT and blockchain in food supply chains," *Journal of Industrial Information Integration*, vol. 25, Art. no. 100261, Jan. 2022.
- [15] M. Malik, A. Hussain, and R. Ahmad, "Authentication challenges in blockchain-based IoT systems," *Sensors*, vol. 22, no. 3, pp. 1–18, 2022.
- [16] T. Fujii, K. Nakamura, and S. Ito, "Lightweight consensus protocols for IoT blockchain networks," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1456–1468, 2024.
- [17] M. Khan, A. Rehman, and S. Ali, "Hybrid blockchain and edge computing for secure IoT in medical applications," *Future Generation Computer Systems*, vol. 145, pp. 234–245, 2025.
- [18] J. Kuang, L. Zhang, and Y. Zhou, "Lattice-based cryptography for blockchain-enabled IoT systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 3, pp. 789–802, 2025.
- [19] Mao, W. (2003) 'A structured operational modelling of the Dolev-Yao threat model', 10th International Workshop on Security Protocols, 2845, pp. 34–46. doi: 10.1007/978-3-540-39871-4_5.
- [20] Li, X., Ma, J. and Moon, S. (2005) 'On the security of the Canetti-Krawczyk model', *Computational Intelligence and Security*, International Conference, 3802 LNAI(90204012), pp. 356–363. doi: 10.1007/11596981_53
- [21] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, et al., "The AVISPA tool for the automated validation of Internet security protocols and applications," in *Proc. Computer Aided Verification*, Edinburgh, U.K., 2005, pp. 281–285. doi: 10.1007/11513988_27.
- [22] Bruno Blanchet, Steve Kremer, and Ben Smyth, "ProVerif: Automatic cryptographic protocol verifier user manual and tutorial," INRIA Research Report, 2023. Available: <https://proverif.inria.fr>.
- [23] S. K. Palit, M. Chakraborty, and S. Chakraborty, "Design, formal verification, and security analysis of SCMChain: A lightweight blockchain-based authentication protocol for IoT-enabled supply chain management," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, pp. 45–62, 2025. doi: 10.1007/s12083-024-01567-9.
- [24] F. Wang, J. Cui, Q. Zhang, H. Zhong, et al., "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial IoT," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 456–470, 2024. doi: 10.1109/TDSC.2023.3245678.
- [25] 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 Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7081–7093, 2020. doi: 10.1109/TII.2020.2971956.
- [26] M. Sidorov, et al., "Lightweight mutual authentication protocol for IoT devices in blockchain-based supply chain," *Journal of Network and Computer Applications*, vol. 135, pp. 62–74, 2019. doi: 10.1016/j.jnca.2019.02.019.