IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2018, accepted December 17, 2018, date of publication January 1, 2019, date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890389

Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains

MICHAIL SIDOROV[1], MING TZE ONG[2], RAVIVARMA VIKNESWAREN SRIDHARAN[3], JUNYA NAKAMURA[1], REN OHMURA[1], AND JING HUEY KHOR[4]

[1]Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan
[2]Department of Engineering Foundation, University of Southampton at Malaysia, Nusajaya 79200, Malaysia
[3]School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.
[4]School of Electronics and Computer Science, University of Southampton at Malaysia, Nusajaya 79200, Malaysia

Corresponding author: Jing Huey Khor (j.khor@soton.ac.uk)

This work was supported in part by the Malaysian Ministry of Higher Education under the Fundamental Research Grant Scheme FRGS/1/2018/ICT04/USMC/02/1.

ABSTRACT Previous research studies mostly focused on enhancing the security of radio frequency identification (RFID) protocols for various RFID applications that rely on a centralized database. However, blockchain technology is quickly emerging as a novel distributed and decentralized alternative that provides higher data protection, reliability, immutability, transparency, and lower management costs compared with a conventional centralized database. These properties make it extremely suitable for integration in a supply chain management system. In order to successfully fuse RFID and blockchain technologies together, a secure method of communication is required between the RFID tagged goods and the blockchain nodes. Therefore, this paper proposes a robust ultra-lightweight mutual authentication RFID protocol that works together with a decentralized database to create a secure blockchain-enabled supply chain management system. Detailed security analysis is performed to prove that the proposed protocol is secure from key disclosure, replay, man-in-the-middle, de-synchronization, and tracking attacks. In addition to that, a formal analysis is conducted using Gong, Needham, and Yahalom logic and automated validation of internet security protocols and applications tool to verify the security of the proposed protocol. The protocol is proven to be efficient with respect to storage, computational, and communication costs. In addition to that, a further step is taken to ensure the robustness of the protocol by analyzing the probability of data collision written to the blockchain.

INDEX TERMS Blockchain, distributed ledger technology, radio frequency identification.

I. INTRODUCTION

With the rapid increase in economic globalization, companies ranging from start-ups all the way to multinational corporations are constantly pushing out products to keep up with the ever growing consumer demand. As a result, supply chains are becoming more convoluted. Many companies have turned to alternative technologies to facilitate tracking and transactions rather than relying on the traditional barcode method. In the early years of this century radio frequency identification (RFID) has proven to be a solution to supply chain product tagging requirements due to its non line of sight detection and simultaneous multiple sensing capabilities [1]. However, RFID technology faces security and privacy threats despite its widespread application and usage. These threats include RFID counterfeiting, sniffing, tracking, denial of service, spoofing, repudiation and replay attacks, etc [2]. Therefore, previous research studies in this area focused mainly on enhancing the security and privacy of various RFID protocols for applications used in conjunction with centralized databases [2]–[8].

Currently most businesses use Enterprise Recourse Planning (ERP) software in conjunction with some additional software to manage the supply chain and they rely on a centralized database for data storage. Thus, the internal system is governed by a single administrator and stored in one location [9]. Through the adoption of cloud computing and Internet of Things (IoT), information stored in the centralized database became accessible from various locations. Although the core disadvantages, e.g. vulnerability to data loss and hacking [10] did not disappear. By looking at the current supply chain management, it is possible to notice a very straightforward operational flow. Once the goods are ready

they are tagged, scanned, and added to the database for the tracking to begin. Although this approach is well adopted, it does not scale and introduces a number of bottlenecks. Not only do analog gaps exist, but synchronizing information e.g. product state between different parties with their own centralized databases or adding more partners to the supply chain ecosystem becomes very difficult. Furthermore, there has to be a degree of trust between parties and someone to account for the shared data. The current approach to supply chain management only provides a limited visibility to where the product was sourced from, where it is at any given moment and with the added desynchronization issue between different parties the same product may appear to be physically located in two geographically different places. Thus, although the supply chain itself has evolved from a simple manufacturer-to-distributor model to a more dynamic ecosystem with multiple parties trying to move products across the supply chain, the underlying system that manages it is heavily outdated. Hence, in order to operate more smoothly, a supply chain is compelled to adopt a method in which products can be traced along every step of the chain, from the supplier, to the manufacturer, all the way to the end user.

A. BLOCKCHAIN TECHNOLOGY FOR THE SUPPLY CHAIN MANAGEMENT

The aforementioned disadvantages can be solved by an emerging technology that plays a vital role not only in enhancing the security for the IoT, but also in providing new management possibilities - blockchain [10], [11]. Introduced in 2008 and initially meant for the purpose of addressing issues with the current economy [12], the technology not only enabled its users to transact without an intermediate third party involved, but has proven to be more versatile than just a method for transferring ownership of wealth [13], [14]. Besides supply chain management, the top most compelling use cases of blockchain infrastructure include digital identity, healthcare, energy market, etc.

The use of blockchain technology for supply chain management has distinct advantages over the currently used approaches. It is a common trend to avoid the middleman as it adds additional overhead expenses, however this entity is extremely useful and was the only way that several parties could reach an agreement on a set of shared data. Blockchain infrastructure in this case not only replaces the middleman, but provides a plethora of new opportunities for supply chain coordination. Thus, we have a system where different businesses agree on a key set of data without an intermediate party having to account for all of the transactions. Furthermore, the core logic of a blockchain dictates that the state of the transaction has to be updated for all of the nodes participating in the network. This enables users to see that the item has changed state or location and that it does not exist simultaneously in two places. The speed of the network depends on the block generation time and can be easily defined. The greater transparency provided by the blockchain can also show where the goods were sourced from and whether they comply with all stated regulations, inherently allowing issues related to counterfeiting to be solved.

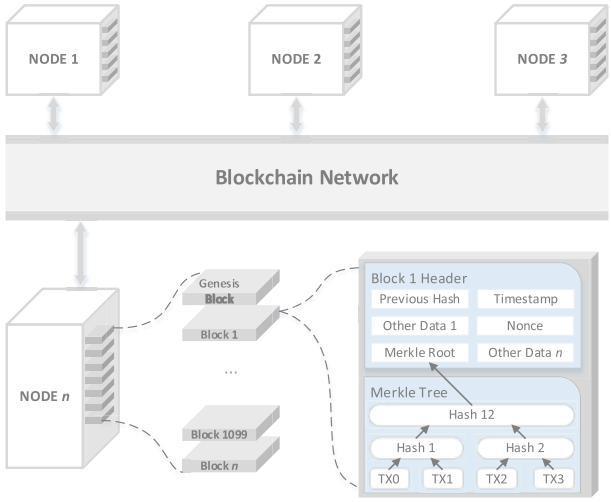


FIGURE 1. Typical blockchain network and simplified block contents.

In simple terms blockchain is an infrastructure with a distributed decentralized ledger (DLT) at its base. Data can only be added to the ledger and not removed [15]. Thus it has a write-once, read mostly (WORM) property. All of the stored data is organized in blocks and a network of peer-to-peer nodes share the copy of the blockchain. Any attempt to tamper with the data will be blocked and a consensus needs to be reached by the majority of participating nodes on the data that is added to the chain. Thus, a single point of failure, compared to a centralized database, does not exist [11]. Consensus methods define the way data is validated. Most widespread ones are Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS) and Byzantine Fault Tolerance (BFT) although other methods exist as well. The sheer amount of redundancy coupled with proven consensus mechanism between participants makes blockchain a very powerful tool for storing and retrieving key pieces of information. By applying this concept to the supply chain we get a revolutionary model where every transaction in between from the source to the retailer is recorded leaving a trail of transparent and immutable history for every participant to see, which practically is not possible when using an old model that relies on a centralized database.

Figure 1 illustrates a typical blockchain network with multiple nodes connected. Furthermore, it shows a simplified version of the block structure. Typical block can be split into two sections: one consisting of a block header that contains timestamp, nonce, previous blocks hash, merkle root, difficulty target and some other useful metadata. The other part is the block body, containing the transactions or data included on the block. This structure, however, is not applicable to all of the existing blockchains as there is a minority that chose to base their structure on a directed-acyclic-graph (DAG).

There are several types of blockchain in existence and they are split into categories according to the node

TABLE 1. Comparison between popular blockchain types and A centralized database.

Description	Public	Permissioned	Private	Centralized Database
Participation	Anyone	Members of organizations	Members of organization	Limited
Write permissions	Granted	Restricted	Restricted	Restricted
Read permissions	Granted	Granted	Restricted	Granted
Speed	Slow	Fast	Fast	Slow
Identity	Anonymous	Anonymous	Known	Known
Security	Impervious to security attacks	Impervious to security attacks	Impervious to security attacks	Vulnerable to security attacks
Transparency	Visible across all supply chain nodes	Visible across all supply chain nodes	Restricted to specific supply chain nodes	Restricted to single supply chain node
Traceability	Yes	Yes	Restricted	No

visibility on the network, how users connect to the network and what rights they get once connected. Thus, we have *public*, *permissioned*and *private* blockchains. Each type of blockchain places a different level of importance on anonymity, immutability, efficiency, and transparency, as shown in Table 1. We can see that a centralized database offers lower security level compared to a blockchain due to the way data is stored, accessed and edited – all in one single location by a single entity with admin rights. Any user with elevated privileges can edit the data by altering a master copy. Security is completely dependent on a local network and is susceptible to a single point of failure.

Public blockchain is accessible to anyone in the world. By using the correct software tools users can join the network, gain equal read and write rights, participate in storing the data and executing the consensus mechanism. The most known and widely used implementations of a public blockchain are Bitcoin, Litecoin, Dash, Monero, Ethereum, etc. [16], [17].

Private blockchain, by definition, has strict permissions to read and write data. A single authority trusted by other users controls transactions being written to the blockchain, and can change the rules of the blockchain if necessary [18]. Read permissions are restricted, as public access to this data might not be always necessary. Doing so allows a much greater room for privacy. Although this approach puts it closer to a centralized database it does add a certain degree of cryptographic auditability.

Permissioned blockchain provides a hybrid between the public and a private blockchain.

Users require approval to join the network and few selected nodes are predetermined to participate in the transaction verification [19]. Each node has the right to grant read or write permission to other nodes on the network and no single node has veto power. It is also possible to have only one entity validating the transactions in which case the consensus mechanism is unnecessary and the network decentralization is maintained by all of the nodes having a copy of the blockchain. Thus, it is impossible for the entity to alter the transaction without any notice.

Not every type of blockchain available can be used for supply chain management without some degree of adjustment, e.g. public blockchain networks are not suitable, as every user will be able to see the information written, which is undesirable. Private ones by default grant full control to a single entity, making it more centralized than necessary. Thus, permissioned blockchain networks with their hybrid features are more suited for use in a supply chain management.

B. POSSIBLE ATTACKS ON A PERMISSIONED BLOCKCHAIN

A number of attacks can be performed on a blockchain. However, the type of attack usually depends on the blockchain type, e.g. a 51% attack that is applicable for a public blockchain is not applicable for private and consortium blockchains due to the nature of how users join the network and the rights they inherit. Thus, there are two attacks that are applicable for a blockchain-enabled supply chain described as follows:

1. Block data modification after the block validation
   This is an attack when the adversary or a network participant tries to change the already stored data in the blockchain. As per Fig. 1 each consecutive block contains the hash of the previous blocks header. This hash is computed from the available metadata (timestamp, previous block header, etc.) before being stored to the next block. Thus this makes it impossible for an attacker to modify just a single block, since the consecutive one has to be modified as well. However, even if multiple blocks are modified, the node is already out of sync with the rest of the network starting the first modified block. Hence, this node is rejected by the network.
2. Sybil attack
   An attacker needs to own a lot of nodes in order to disrupt the processing of valid transactions. Since all nodes need to request permission to join the permissioned blockchain, it is therefore difficult for an attacker to register as a distributor or a retailer node. Attacker might be able to register as end user node, however end user is granted with read access only. Therefore, the attacker would be unable to flood the network with a large number of fraudulent transactions.

C. RELATED WORK

Although blockchain offers a lot of benefits for the supply chain, there are a number of barriers to its widespread adoption. One of the barriers is the security and privacy issues associated with the integration of RFID technology in the blockchain system. As blockchains are in their infancy, there is limited research conducted in this area.

Toyoda *et al.* [20] proposed a novel blockchain-based product ownership management system of RFID-attached products for anti-counterfeits to use in the post supply chain. A full-fledged protocol was designed to enable each party, including supply chain partners and customers, to transfer and prove the ownership of RFID tag-attached products based on electronic product code (EPC). However, the EPC is sent as a fixed value during the whole process. Hence, an adversary can easily conduct a tracking attack to monitor the movement of the RFID tag-attached products based on this value.

Lightweight RFID authentication protocols have been researched extensively [2]–[8]. However, security issues still exist. For example, Tewari and Gupta [21] proposed a secure ultra-lightweight mutual authentication protocol for RFID use in IoT networks. The protocol uses two bitwise operations, including the XOR and left rotation to provide data protection and achieve efficient utilization of storage and communication resources. The protocol claimed to provide a full spectrum of security features e.g. mutual authentication, confidentiality, integrity, anonymity, forward secrecy, as well as security against main-in-the-middle, replay, de-synchronization, and disclosure attacks. However, this protocol was eventually shown to be susceptible to full disclosure, man-in-the-middle (MITM), tracking, and de-synchronization (De-Sync) attacks [22].

A cloud-based mutual authentication protocol for RFID tags used in supply chain system was proposed by Lin *et al.* [23]. Protocol utilized XOR operations and a hash function as its core and was claimed to achieve confidentiality, untraceability, mutual authentication, and forward secrecy. Furthermore, it was claimed that the protocol is resistant to tag/reader impersonation attacks, replay attacks, desynchronization attacks, and denial of service (DoS) attacks. However, the protocol was later proven to be vulnerable to de-synchronization and DoS attacks [24].

A key management identity authentication (KMIA) protocol for high throughput RFID system was proposed by Hsu *et al.* [25]. This protocol was claimed to be able to achieve secure mutual authentication and data secrecy. However, the KMIA protocol is susceptible to three attacks, namely the man-in-the-middle attack, denial-of-service attacks and replay attacks because of its design flaws. An attacker can perform man-in-the-middle attacks by modifying the E_K (R) message being sent from a legitimate tag to a reader. In addition, an attacker can perform DoS attacks by blocking the E_K (R) message that is sent from a legitimate tag to a backend server via a reader. Furthermore, it is possible to perform a replay attack by capturing the E_K (TID $\oplus$ K) and E_K (R) messages sent from a legitimate tag to a legitimate reader during a specific session.

Mujahid *et al.* [26] introduced a new ultra-lightweight primitive, namely the pseudo-Kasami code. It was claimed that it helped to achieve secrecy for RFID systems by making the secret keys unpredictable. Authors also proposed a mutual authentication RFID protocol (KMAP) using the pseudo Kasami-code, XOR, hamming weight and bitwise rotation operations. However, this protocol was later proven to be susceptible to a de-synchronization attack [27]. Therefore, Mujahid *et al.* [2] proposed an improved version of the protocol called KMAP+ to remove the flaws of the initially proposed protocol.

Some of the researchers proposed to use a secure key distribution for RFID-enabled supply chain to combat tracking attacks. Juels *et al.* [28] proposed secret-sharing across space and secret-sharing across time approaches to provide data privacy protection. For secret-sharing across space, the secret key is distributed across a set of tags and will be received by the supply chain at the same time. On the other hand, for secret-sharing across time, the distributed secret key will arrive at staggered times in the supply chain. These two approaches have been shown to be susceptible to tracking attacks because a tag always replies with the same constant message. Proposals were then made to solve the tracking issues found in the Juels *et al.*'s [28] approach, e.g. Cai *et al.* [29] proposed a secure key distribution protocol that ensures secure ownership transfer of tags in supply chain without any possibility of executing a tracking attack. However, in order to achieve this security, the protocol requires a special tag that is capable of supporting computationally heavy hash value calculation, as opposed to computationally light bitwise operations. Li *et al.* [30] proposed a resilient secret sharing scheme for key distribution that focused on any pair of consecutive parties only. This protocol similarly was designed requiring the use of hash function to ensure the integrity of the transmitted message, which is computationally costly. Toyoda and Sasase [31] proposed a secret sharing scheme which requires to use a large number of dummy tags to solve the tracking issue faced by the approach in [28]. However, this solution has one obvious downside, where additional cost has to be incurred to purchase the extra dummy tags to be later added to the supply chains. The latest RFID protocol, Gen2V2 [32] provides an additional security feature to the previous protocol, Gen2V1. This feature known as Untraceable command, which enables a tag to expose its secret information, including EPC, TID, and user memory to privileged readers only. However, this security feature is susceptible to security attacks because the Gen2V2 protocol does not guarantee that any malicious reader complying with Gen2V2 protocol can set itself as a privileged reader and undo the untraceable feature of a tag.

Therefore, there is a need for a robust and efficient RFID protocol that eliminates any security flaws still found in the previous ones for integration with blockchain infrastructure.

D. MAIN CONTRIBUTIONS

The main objective of this paper is to propose an ultralightweight mutual authentication RFID protocol for integration in a blockchain enabled supply chain. The integration itself is a novel approach and the proposed protocol provides full security protection against possible attacks such as key disclosure, replay, main-in-the-middle,

desynchronization, and tracking, ensuring the security of the data being written to the blockchain network. The added visibility and transparency provided by the resulting fusion of proposed blockchain enabled supply chains is able to facilitate the tracing of products across supply chain nodes and prevent counterfeiting.

The following are the main contributions of this paper:

1. A supply chain system model with different access and consensus levels for each of the chain nodes is proposed. Each access level is represented by using odd or even hamming weight value for simplicity.
2. Until present, there was no research done on designing lightweight RFID protocols suitable for a decentralized database. Hence, a new protocol that can be used in a blockchain for supply chain system is proposed. This protocol adds the necessary protection layer which is crucial to enable a secure data transmission over a communication channel. Hence, protecting the resource constrained RFID tags from key disclosure, replay, man-in-the-middle, de-synchonization, and tracking attacks.
3. Encrypted data is stored in a permissioned blockchain, visible and accessible to every party involved in the supply chain, unlike traditional supply chain systems that use centralized database.
4. This protocol solution is applicable to be used with any blockchain by specifying the access levels of participating nodes accordingly.
5. The proposed protocol is designed with computational cost in mind. The use of ultralightweight bitwise operation composed of exclusive-OR, hamming weight, and rotation operations allowed us to minimize the computational cost at the RFID tag side.
6. The probability of data collision in a permissioned blockchain is analyzed to validate the robustness of the proposed protocol.
7. The widely used GNY logic and AVISPA simulation tool is used to formally verify the security of the proposed protocol. The general security analysis further shows that the proposed protocol is fully secured against various known attacks.

E. PAPER STRUCTURE

The remaining of this paper is structured as follows:

- Section II describes the designed ultra-lightweight protocol and provides a clear executional example
- Section III analyzes the data collision probability in a blockchain of the proposed protocol
- Section IV and V show general and formal analyses of the proposed protocol
- Section VI presents the simulation results obtained using AVISPA tool
- Section VII illustrates performance of the designed protocol and compares it to the state of the art proposals
- Section VIII concludes the paper

TABLE 2. Access and consensus level of supply chain nodes.

Supply chain nodes	Access	Consensus
Manufacturer	Write and Read	Yes
Distributor	Write and Read	No
Retailer	Write and Read	No
End user	Read only	No

II. ULTRALIGHTWEIGHT PROTOCOL FOR BLOCKCHAIN ENABLED SUPPLY CHAINS

In this paper, permissioned blockchain network is used for the supply chain management due to the higher levels of privacy, security and scalability it can provide. In order to access a permissioned blockchain participants need authorization. In some cases this will include different permission levels, such as read only, read and write. These authorizations are granted by the network members or governing body [19]. Thus, a certain level of relationship and trust is assumed between participants.

Main supply chain nodes are: the manufacturer, distributor, retailer, and end user. Table 2 shows access and consensus levels assigned to each of the supply chain nodes. The manufacturer node plays a vital role as a governing body to control the supply chain platform. It has full access and is capable of granting permissions to certain parties to join the network. In addition, it is the only node that can execute a consensus protocol in this network. The rest of the nodes in the network, such as the distributor and retailer have the same full access control as the manufacturer node, but, they are unable to validate transactions. On the other hand, end user nodes are only allowed to read from the network. Since different supply chain nodes have different levels of access, the tag needs to distinguish between them and either update or keep its secret data intact accordingly. RFID tag is computationally constrained, thus, a simple solution is introduced to notify of the access level and consists of sending a specific value to the tag. In this paper, an even hamming weight of random number is used to represent a supply chain node that has both read and write access, whereas an odd hamming weight of random number is used to represent the end user node that only has read access.

The protocol involves the following parties: tag, reader, and supply chain node. The latter one consists of the manufacturer, distributor, retailer, and an end user. Since low cost RFID tag is computationally constrained, the proposed protocol is designed using bitwise XOR and rotation operations. The supply chain node, however, has no computational constrains. Therefore, it can perform more demanding computational operations, such as generating SHA-256 hash function and product history verification based on the data stored in the blockchain. SHA-256 hash function is used to encrypt the secret data of IDS and K before storing it permanently in the header of a block. These stored values can

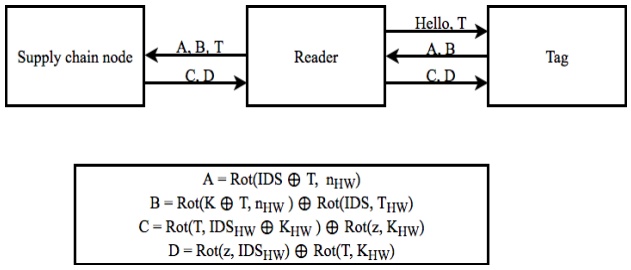


FIGURE 2. Authentication phase of an ultralightweight protocol.

TABLE 3. Notation used in the proposed protocol.

Notation	Description
$\oplus$	Exclusive-OR
T	Random number generated by reader
n	Random number generated by tag
z	Random number generated by supply chain node
$Rot(X, Y_{HW})$	Left rotate of X by hamming weight of Y
$RRot(X, Y_{HW})$	Right rotate of X by hamming weight of X
IDS_{new}	Current session pseudonym
IDS_{old}	Previous session pseudonym
K_{new}	Current session secret key
K_{old}	Previous session secret key
$hash$	SHA-256

then be used during the authentication process for all supply chain nodes.

An assumption is made where the communication channel between supply chain node and reader is secure, whereas the communication channel between the reader and a tag is insecure. In addition, data stored in the blockchain is assumed to be secure due to the nature of a distributed ledger. Therefore, the protocol is designed to protect transmitted messages over the communication channel between the reader and the tag. Figure 2 summarizes operations that take place during the authentication phase between the supply chain node, reader, and the tag, followed by a comprehensive explanation of each step. Notations being used in the proposed protocol are described in Table 3.

The description of the proposed ultralightweight protocol is as follows:

1. To initiate a session, the reader sends a *Hello* message and a random number, T to the tag.
2. After receiving both of the messages the tag computes messages A and B by using its stored IDS, K, generated random number n, as well as received random number T. The tag sends the computed A and B to the reader.
3. The reader forwards messages of T, A, and B to the supply chain node.
4. The supply chain node extracts IDS and K by generating n'_{HW} and performs the following operations until a matched $hash(IDS||K)$ is obtained from permissioned blockchain. Since $n$is 96 bits, we have n_{HW} between 0 and 96.

$$IDS' = RRot\left(A, n'_{HW}\right) \oplus T$$
$$K' = RRot(B \oplus Rot\left(IDS', T_{HW}\right), n'_{HW}) \oplus T$$

Based on the $hash(IDS||K)$, the supply chain node can check and track the product history together with permission level by reading the data from the blockchain. If the product has a correct history record in terms of ownership, timestamp, location, and product status, the supply chain node can authenticate the tag. Next, supply chain node generates a random number, z. For write and read permission level, an even hamming weight of random number is generated; otherwise, an odd hamming weight of random number is created. The supply chain node computes C and D and sends those messages to the reader. Next, the supply chain node of the manufacturer, distributor, or retailer updates IDS_{new} and K_{new} accordingly. After this update and product history verification step, the supply chain adds a transaction with the new $hash(IDS_{new}||K_{new})$ and previous $hash(IDS||K)$ values to the blockchain. However, for end user supply chain node no data is updated in the blockchain, since this node is only granted with read access.

5. The reader forwards messages C and D to the tag.
6. After receiving messages C and D, the tag extracts random number z' from the received message D.

$$z' = RRot\left((D \oplus Rot\left(T, K_{HW}\right)\right), IDS_{HW})$$

The tag authenticates the reader if message C', which is computed from the extracted z' is equal to the received C. After the authentication, the tag updates its IDS_{new} and K_{new} if the hamming weight of z' is an even value. If the hamming weight of z' is an odd value, the tag will not update its IDS_{new} and K_{new}.

$$IDS_{new} = Rot(K \oplus n_{HW}, IDS_{HW}) \oplus Rot(IDS \oplus K, T_{HW})$$
$$K_{new} = Rot(K, T_{HW}) \oplus Rot(T \oplus n_{HW}, K_{HW})$$

A concrete example that illustrates how the mutual authentication protocol is executed when RFID devices communicate within the supply chain is described as follows. Firstly, distributor, retailer, and end user nodes need to register and clarify their identities through the manufacturer node, and their identity (ID) will then be assigned with a 256 bit string. Since a typical ID is quite long, an abbreviated form of a 256 bit string, e.g. EO8.....768, will be used in the sections below. A practical scenario of the protocol functionality during the execution of an authentication phase between supply chain nodes and RFID tags is shown in Figure 3. RFID readers are not shown in this figure as their function is mainly to forward messages between RFID tags and supply chain nodes. Assume a distributor node with an ID value of EO8....76F ships out a product, the RFID tag attached to the

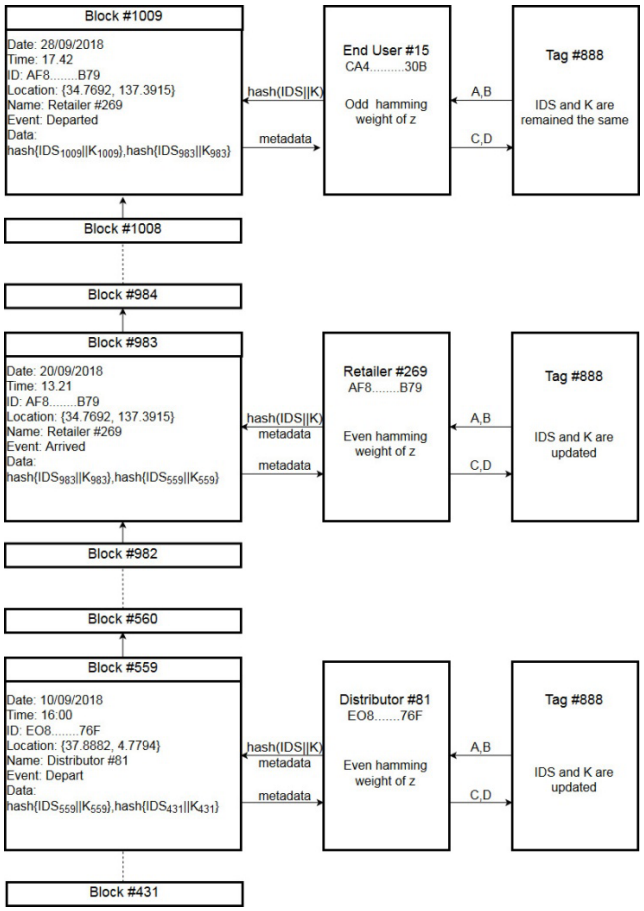


FIGURE 3. Practical scenarios of mutual authentication protocol in blockchain-enabled supply chain.

product will be scanned using the distributor node's reader. The authentication process between the RFID tag and distributor node begins by comparing the computed $hash(IDS||K)$ value and the one stored in the blockchain for a match. Suppose a matching $hash(IDS||K)$ value is found in block 431, then the distributor node which has both read and write access updates its IDS and K values and stores both the new and old computed $hash(IDS||K)$ values as well as other supply chain metadata in a block. For this example, suppose the block number is 559. Next, the distributor node generates a random number z which must have an even hamming weight, this is then used to compute messages C and D which are then send to the tag via the reader. The tag then authenticates the distributor node and since the value of z extracted from the C and D messages has an even hamming weight, it then updates its IDS and K values.

Suppose the product arrives at a retailer with an ID value of AF8...B79, the product is then scanned and its product history can be traced based on $hash(IDS||K)$ values, which can be found in the block with number 559. The product history can be traced back further based on its previous $hash(IDS||K)$ value that is stored in block 559, which also can be found in block 431. After authenticating the tag, the retailer node updates its IDS and K, and stores new and old versions of the $hash(IDS||K)$ as well as other supply chain metadata in the 983rd block. The tag then authenticates the retailer node and since the extracted value of z is even hamming weight, it updates its IDS and K values. Assuming at some point the product is sold to an end-user with an ID value of CA4..... 30B which is reflected in the block number 1009, the end user can trace the entire history of the product in the manner similar to the way in which the retailer can. An important difference in the mutual authentication process between an end user node and an RFID tag as opposed to other supply chain nodes and an RFID tag is that the end user node does not add a block to the blockchain, since the end user node is granted with read only access.

III. COLLISION ANALYSIS OF THE PROPOSED PROTOCOL

In this section the robustness of the proposed protocol is analyzed. Recall that typically every supply chain node will have a database storing all the tags created so far. As every tag is uniquely identified by a (IDS, K) pair, where IDS and K are distinct bit strings each of length$n$. The supply chain database will have a list of j, (IDS, K) bit string pairs corresponding to all the j tags which have been created and are in the blockchain.

A reader would typically be communicating with a tag, and needs to be certain of two things. Firstly, that it is communicating with a legitimate tag, and not a rogue one, and secondly, that it is communicating with a legitimate tag whose messages have not been intercepted maliciously or otherwise corrupted.

As part of the authentication process the reader needs to send three n bit length strings to the supply chain node for verification. This is needed to be certain that the reader is indeed communicating with a legitimate tag. Of the three bit strings that the reader sends to the supply chain node, two are passed on from the tag (bit strings A and B) and one from the reader itself (bit string T). The bit strings A and B are calculated by the tag using the bit string T received from the reader when communication between the two entities was first established, according to following:

$$A = Rot\,(IDS \oplus T, n_{HW}) \tag{1}$$

and

$$B = Rot\,(K \oplus T, n_{HW}) \oplus Rot\,(IDS, T_{HW})\,(2) \tag{2}$$

where n_{HW} is a tag generated random hamming weight, that takes values between 1 and n. The supply chain node only receives bit strings A, B, and T. It has to work out the matching n_{HW} value and authenticate the tag that is communicating with the reader. Since the supply chain node does not receive the tag generated hamming weight n_{HW}, to determine the (IDS, K) pair which corresponds to the received bit strings A and B, it has to, in the worst case scenario perform the reverse operation in (1) and (2) above, using n trial hamming weight values, denoted by n'_{HW}, in place of the true hamming

weight, n_{HW}, which was used by the communicating tag in its calculation of A and B. That is to say, the supply chain node needs to, in the worst case scenario, calculate in sequence, n pairs of trial (IDS, K) tag pairs denoted by (IDS', K'), each time checking against its list of already stored$k$ (IDS, K) tag pairs for a match according to the following:

$$IDS^{'} = RRot\left(A, n_{HW}^{'}\right) \oplus T \tag{3}$$

and

$$K^{'} = RRot\left(B \oplus Rot(IDS^{'}, T_{HW}), n_{HW}{}^{'}\right) \oplus T \tag{4}$$

A potential problem arises in that the trial (IDS', K') pair computed using a trial hamming weight value n'_{HW} might match a wrong (IDS, K) pair already in the list of k, (IDS, K) pairs stored in the permissioned blockchain. This is known as a collision problem and the probability of it needs to be determined. The probability of such a collision with a wrong (IDS, K) tag pair is denoted as $\left\{\left(IDS^{'}, K^{'}\right) = (IDS_{incorrect}, K_{incorrect})\right\}$, $(IDS_{incorrect}, K_{incorrect})$ is one of the k tag pairs stored in the blockchain, but does not correspond to the bit string A and B, i.e., A and B were not computed from $IDS_{incorrect}$ and $K_{incorrect}$.

A feature of the protocol is that such a probability can be easily calculated and is equal to finding the probability that if an XOR operation is performed on two distinct random bit strings, X_1 and X_2, each of length n, the result matches a third random distinct bit string, $X3$, of length n. That is to say the probability

$$P\left(X_1 \oplus X_2 = X_3\right) \tag{5}$$

needs to be known.

Since $X_1 \oplus X_2 = X_3$ if and only if

$$X_{1,i} \oplus X_{2,i} = X_{3,i} \tag{6}$$

for all $1 \leq i \leq n$, where $X_{1,i}$, $X_{2,i}$, and $X_{3,i}$ denote the i^{th} bits in the bit strings X_1, X_2, and X_3 respectively.

Noting that $X_{3,i}$ is either a 0 or 1 with equal probability, then

$$\begin{aligned} &P\left(X_{1,i} \oplus X_{2,i} = X_{3,i}\right) \\ &\quad = P\left(X_{3,i} = 0\right).P\left(X_{1,i} \oplus X_{2,i} = 0\right) \\ &\quad + P\left(X_{3,i} = 1\right).P\left(X_{1,i} \oplus X_{2,i} = 1\right) \end{aligned} \tag{7}$$

But $P\left(X_{1,i} \oplus X_{2,i} = 0\right) = \frac{1}{2}$ and likewise, $(X_{1,i} \oplus X_{2,i} = 1) = \frac{1}{2}$. Using the values of $P(X_{1,i} \oplus X_{2,i} = 0)$ and $P(X_{1,i} \oplus X_{2,i} = 1)$ in (7), the following is obtained $P(X_{1,i} \oplus X_{2,i} = X_{3,i}) = \frac{1}{2}$.

From (6), the following occurs naturally

$$\begin{aligned} P\left(X_1 \oplus X_2 = X_3\right) &= \prod_{i=1}^{n} P(X_{1,i} \oplus X_{2,i} = X_{3,i}) \\ &= \left(\frac{1}{2}\right)^n \end{aligned} \tag{8}$$

(8) now allows us to find the probability that the trial (IDS', K') pair calculated in (3) and (4) above matches a wrong (IDS, K) pair stored in the blockchain to be

$$\begin{aligned} &P\{\left(IDS^{'}, K^{'}\right) \\ &\quad = (IDS_{incorrect}, K_{incorrect})\} \\ &\quad = P\left(IDS^{'} = IDS_{incorrect}\right).P(K^{'} = K_{incorrect}) \\ &\quad = P\left(RRot\left(A, n_{HW}^{'}\right) \oplus T = IDS_{incorrect}\right) \\ &P(RRot(B \oplus Rot\left(IDS^{'}, T_{HW}\right), n_{HW}^{'}) \\ &\quad \oplus T = K_{incorrect}) \\ &\quad = \left(\frac{1}{2}\right)^n \cdot \left(\frac{1}{2}\right)^n = \frac{1}{4^n} \end{aligned} \tag{9}$$

since $RRot(A, n_{HW}^{'}) \oplus T$ and RRot(B $\oplus$ $Rot\left(IDS^{'}, T_{HW}\right)$, $n_{HW}^{'}$) and T are all random bit strings of length n.

Since in the worst case scenario the supply chain node needs to try n trial hamming weight values $n_{HW}^{'}$ and there are k number of (IDS, K) pairs already in the permissioned blockchain, the probability that the supply chain node wrongly identifies the node which is attempting communication with it, denoted by $P_{incorrect_{match}}$ must satisfy

$$P_{incorrect\_match} \leq \frac{(n-1)(k-1)}{4^n} \tag{10}$$

IV. SECURITY ANALYSIS OF THE PROPOSED PROTOCOL

The following assumptions are used in the analysis of five possible attacks on the proposed protocol:

1. An adversary has the capability to initiate communication with the reader and a tag.
2. An adversary is able to eavesdrop, intercept, block, and modify messages sent during communication between the reader and a tag.

A. KEY DISCLOSURE

Key disclosure attack is an attack where adversary can decrypt transmitted messages by obtaining secret information that was used to encrypt the message. The protocol, however, is secured against a key disclosure attack. Secret information IDS and K is hashed before being written to the blockchain. Hash function is a one-way function that is irreversible. For a communication channel between the reader and a tag, K and IDS are encrypted with new random numbers T and n for each new session. Although T is sent as a plaintext from the reader to the tag, the n value, however, is not. A threshold of five is set. Thus, limiting the number of attempts the tag can send messages A and B to the reader within a certain period of time. If the number of attempts exceeds the threshold, the reader sends a $KILL$ command to terminate the tag. In addition, the reader terminates the current session and initiates a new session with new random number, T. Hence, the tag computes messages A and B with the new random number T and n. As a result, the adversary is unable to perform brute force attack to obtain IDS and K because of the limited number of trials.

Furthermore, the adversary is unable to guess the values of *IDS* and *K* based solely on the messages *C* and *D*, as they are encrypted with a new random number *z*.

B. REPLAY ATTACK

Replay attack occurs when an adversary records messages that are exchanged in a communication channel and replays them to steal information or gain access. Several scenarios are described below how an adversary may attempt to perform a replay attack and fail to do so.

1. Adversary replays messages *A* and *B* captured during the previous session to the reader. The adversary however, will not be successful, since the reader is unable to authenticate the messages as being sent by a genuine tag because *IDS* and *K* are encrypted with new random numbers (*T* and *n*) for every new session. This will result in the reader computing *IDS* and *K* values which are different from the ones stored in the blockchain.
2. Adversary replays messages *C* and *D* captured in a previous session to the tag. Again the adversary will not be successful since the tag is unable to authenticate the messages since the computation of messages *C* and *D* requires different random numbers (*T* and *z*) for each new session.

Since the adversary is unable to obtain any messages from reader and tag using either, or both, of the above malicious schemes, the proposed protocol is able to resist replay attacks.

C. MAN-IN-THE-MIDDLE ATTACK

Man-in-the-middle attack is an attack that happens during a signal transmission where adversary eavesdrops, intercepts and manipulates the information. Adversary may attempt to perform a man-in-the-middle attack using several approaches described below, all of which will be unsuccessful:

1. Adversary blocks messages *A* and *B*, and then modifies them before sending to the reader. The supply chain node is unable to obtain a matching $hash(IDS||K)$ value in the blockchain because the adversary is unable to obtain the exact values of *n*, *K*, and *IDS* to compute messages *A* and *B*.
2. Adversary blocks messages *C* and *D*, then modifies them before sending to the tag. Since messages *C* and *D* are computed from updated random number *z*, *K*, *IDS* for each session, the adversary is unable to guess correct messages of *C* and *D*. Therefore, the tag is unable to authenticate the adversary because the messages C' and D' computed are different compared to the modified messages *C* and *D*.

This proves that the proposed protocol is secure from man-in-the-middle attacks.

D. DE-SYNCHRONIZATION ATTACK

Secret key de-synchronization is a typical RFID threat where an adversary blocks communication channel between the reader and a tag and maliciously changes the secret data stored in the database and the tag. In this protocol, both supply chain node and tag update their secret information, *IDS* and *K* at the end of each successful session to maintain their synchronization. An adversary may apply several approaches to de-synchronize the secret information between the supply chain node and a tag, all of which will be unsuccessful:

3. Adversary blocks messages *A* and *B* from reaching the reader. Since the reader does not receive messages *A* and *B,* it will keep waiting to receive message from the tag for a certain period of time, and then it will terminate the current session.
4. Adversary blocks messages *C* and *D* from reaching the tag. This causes the secret key and *IDS* stored in the tag to be different than the ones stored in the blockchain. Therefore, *IDS* and *K* values used by the tag to compute messages *A* and *B* are old versions of *IDS* and *K* stored in the blockchain. Since there are two versions of $hash(IDS||K)$ stored in a block, if the $hash(IDS||K)$ belongs to an old version of the $hash(IDS||K)$ value, a new transaction block with a new $hash(IDS||K)$ will be added to the blockchain. Therefore, the secret information stored in both the blockchain and the tag are synchronized. This allows a new transaction block to be added to the blockchain in the future. The block that has a different $hash(IDS||K)$ becomes orphaned block.
5. Adversary may attempt to modify messages *C* and *D* in order to change the random number *z* either to a number with an even or odd hamming weight. By changing the value of *z* the true access level of the supply chain node will be misrepresented. The adversary might hope that this would cause a tag which receives a wrong even hamming weight number to wrongly updates its *IDS* and *K* value, or cause a tag which receives a wrong odd hamming weight number to not update its *IDS* and *K* value. However, since the protocol always requires the integrity of *z* to be verified using message *C* at the tag side, the tag will not authenticate the information sent by the adversary as being genuine, because the computed C' and received C' are different.

This proves that the proposed protocol is secure from de-synchronization attack.

E. TRACKING ATTACK

This kind of attack is aimed at tracking the movement of a tag based on a constant response returned by the tag to the readers queries. This constant response can be prevented by encrypting it with a secret key. In this protocol, messages *A*, *B*, *C*, and *D* are encrypted with secret information *IDS* and *K*, as well as random numbers *T, n* and *z*. Both *IDS* and *K* are updated at the end of each successful session using random numbers *T* and *n* for each session.

V. FORMAL ANALYSIS OF THE PROPOSED PROTOCOL

To analyze the security correctness of the protocol, GNY logic is used. It is more complex compared to Burrows-Abadi-Needham logic due to the fact that it has several

new and improved rules [33]. Some important notation of GNY logic that is used in this paper is illustrated below:

- $P \triangleleft Q$: *P has seen Q*
- $P \triangleleft *Q$: *P has seen Q, where Q did not originated from P*
- $P \ni Q$: *P possesses Q*
- $P \mid \equiv Q$: *P believes Q*
- $P \mid \sim Q$: *P once said Q*
- $P \mid \equiv \phi(X)$: *P recognize X*
- $P \mid \equiv \#(X)$: *P believes X is fresh*
- $\{X\}_K$: *X is symmetrically encrypted with K*
- *(X,Y): X or Y is part of (X,Y)*
- *H(X): one-way function of X*
- $P \overset{K}{\leftrightarrow} Q$: *K is shared between P and Q*

The following shows the logical postulates of the GNY logic being used in this paper:

- *I1:* $\frac{P \triangleleft *\{X\}_K, P \ni K, P \mid \equiv P \overset{K}{\leftrightarrow} Q, P \mid \equiv \phi(X), P \mid \equiv \#(X,K)}{P \mid \equiv Q \mid \sim X, P \mid \equiv Q \mid \sim \{X\}_K, P \mid \equiv Q \ni K}$
- *T1:* $\frac{P \triangleleft *X}{P \triangleleft X}$
- *P1:* $\frac{P \triangleleft X}{P \ni X}$
- *R6:* $\frac{P \ni H(X)}{P \mid \equiv \phi(X)}$
- *F1:* $\frac{P \mid \equiv \#(X),}{P \mid \equiv \#(X,Y), P \mid \equiv \#(F(X))}$
- *F11:* $\frac{P \mid \equiv \#H(X), P \ni H(X)}{P \mid \equiv \#(X)}$

Formalized messages (M) delivered between reader and tag can be obtained based on the authentication phase of the protocol. A formalized version of the protocol is shown below:

- *M1: Reader -> Tag: Hello, T*
- *M2:Tag->Reader:* $Rot\,(IDS \oplus T, n_{HW})\,, Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- *M3: Reader->Suppy chain node: T,* $Rot(IDS \oplus T, n_{HW}), Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$

M4: Supply chain node-> Reader: $Rot(T, IDS_{HW} \oplus K_{HW}) \oplus Rot(z, K_{HW}), Rot(z, IDS_{HW}) \oplus Rot(T, K_{HW})$

- *M5: Reader->Tag:* $Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})\,, Rot(z, IDS_{HW}) \oplus Rot(T, K_{HW})$

Based on the GNY logic formulas, the protocol messages can be idealized as seen below. Plaintexts such as *Hello* and T are omitted from the protocol messages at this stage.

- *IM1: Reader* $\triangleleft *Rot\,(IDS \oplus T, n_{HW})\,, *Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- *IM2: Suppychainnode* $\triangleleft *Rot\,(IDS \oplus T, n_{HW})\,, *Rot\,(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- *IM3: Reader* $\triangleleft * Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})\,, Rot(z, IDS_{HW}) \oplus Rot(T, K_{HW})$
- *IM4: Tag* $\triangleleft * Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})\,, Rot(z, IDS_{HW}) \oplus Rot(T, K_{HW})$

The initial assumptions of the protocol are represented using GNY logic formulas. These assumptions specify initial possess and belief of data between reader and tag.

- *A1: Reader* $\mid \equiv \#(T)$
- *A2: Tag* $\mid \equiv \#(n)$
- *A3: Reader* $\mid \equiv \#(z)$
- *A4: Tag* $\mid \equiv Tag \overset{K}{\leftrightarrow} Reader$
- *A5: Reader* $\mid \equiv Reader \overset{K}{\leftrightarrow} Tag$
- *A6: Tag* $\mid \equiv Tag \overset{IDS}{\leftrightarrow} Reader$
- *A7: Reader* $\mid \equiv Reader \overset{IDS}{\leftrightarrow} Tag$
- *A8: Tag* $\ni K$
- *A9: Reader* $\ni K$

The objective of the protocol is to guarantee that fresh messages are sent from trustable entities. The goals of the protocol are shown below:

- G_1: Reader $\mid \equiv$ Tag $\mid \sim Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- G_2: Tag $\mid \equiv$ Reader $\mid \sim Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$

G_1 means that the reader believes the tag has sent message $B, Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$. This indicates that the adversary has not changed the message B, which was computed by the tag and sent to the reader.

G_2 means that the tag believes the reader has sent message C, $Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$. This indicates that the adversary has not changed the message C, which was received from the supply chain node and sent to the tag.

In order to show that the protocol provides secure mutual authentication between reader and tag, *I1* is used to prove *G1*, $Reader \mid \equiv Tag \mid \sim Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$, where the specified conditions must hold:

- *C1:* $Reader \triangleleft *Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- *C2:* $Reader \ni K$
- *C3:* $Reader \mid \equiv Tag \overset{K}{\leftrightarrow} Reader$
- *C4:* $Reader \mid \equiv \phi Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$
- *C5:* $Reader \mid \equiv \#Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$

By using GNY logic, we can prove that if any of the above conditions do not hold, the protocol can be considered insecure.

C1 is obtained from *IM1*, *C2* is obtained from *A9*, and *C3* is obtained from *A4*, respectively.

D1 is obtained by applying *T1* to *C1*.

$$Reader \triangleleft Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$$

D2 is obtained by applying *P1* to *D1*.

$$Reader \ni Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$$

C4 is obtained by applying *R6* to *D2*.

$$Reader \mid \equiv \phi Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$$

C5 is obtained by applying *F1* to *A1*.

$$Reader \mid \equiv \#Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$$

Therefore, the goal, *G1* is achieved by applying*I1* to *C1*, *C2*, *C3*, *C4*, and *C5*.

Reader $\mid \equiv$ Tag$\mid \sim Rot(K \oplus T, n_{HW}) \oplus Rot(IDS, T_{HW})$

To prove G2: Tag $\mid \equiv$ Reader $\mid \sim Rot(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$,*I1* is applied and the following conditions must hold:

- *C6:Tag* $\triangleleft * Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$
- *C7: Tag* $\ni K$
- *C8: Tag* $| \equiv Reader \overset{K}{\leftrightarrow} Tag$
- *C9: Tag* $| \equiv \phi Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$
- *C10: Tag* $| \equiv \#Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$

C6 is obtained from *IM4*, *C7* is obtained from *A8*, and *C8* is obtained from *A6*, respectively.

D3 is obtained by applying *T1* to *C6*.

$$Tag \triangleleft Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$$

D4 is obtained by applying *P1* to *D3*.

$$Tag \ni Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$$

C9 is obtained by applying *R6* to *D4*.

$$Tag| \equiv \phi Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$$

C10 is obtained by applying *F1* to *A3*.

$$Tag| \equiv \#Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$$

Therefore, the goal, *G2* is achieved by applying *I1* to *C6*, *C7*, *C8*, *C9*, and *C10*.

$Tag \mid \equiv Reader| \sim Rot\,(T, IDS_{HW} \oplus K_{HW}) \oplus Rot\,(z, K_{HW})$

VI. SIMULATION FOR FORMAL ANALYSIS USING AVISPA TOOL

The proposed protocol is coded using High Level Protocol Specification Language (HLPSL) in order to be formally verified using a broadly accepted formal verification tool, namely AVISPA [34]–[36]. The tool consists of four backends listed as follows:

1. On-the-fly-Model-Checker (OFMC)
2. Constraint Logic based Attack Searcher (CL-AtSe)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)

OFMC and CL-AtSe backends have been selected for formal security verification. These two backends can support the implementation of the exclusive-OR operation used in the proposed protocol. Details of the HLPSL specification and AVISPA tool can be found in [35].

A Dolev-Yao model check of the protocol together with a resistance check against a replay attack is conducted. In the Dolev-Yao model check, the OFMC and CL-AtSe backends check for possible man-in-the middle attacks by an intruder in the system. For the replay attack resistance verification, the OFMC and CL-AtSe backends check whether those attacks are possible when legitimate entities are executing the specified protocol with intruders present within the system. Detailed descriptions of these verifications are given in [35].

The simulated results using the OFMC and CL-AtSe backends are shown in Figure 4. As shown in the OFMC simulation result, the depth of the search is 5, where 24 nodes have been searched in 0.22 seconds. On the other hand, the CL-AtSe backend result shows that 6 states were analyzed,

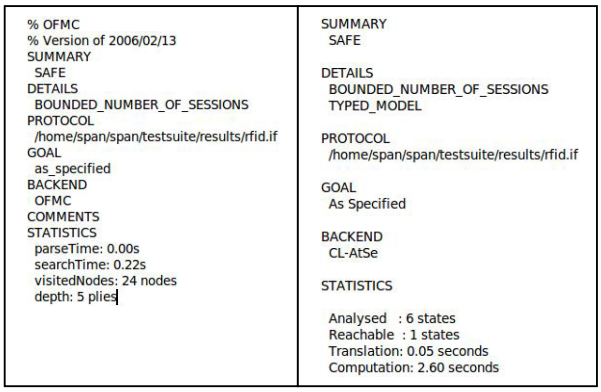


FIGURE 4. Simulation results using OFMC and CL-AtSe backends.

where only one state was reachable, taking 0.05 seconds for translation and 2.60 seconds for computation. The summary result clearly states that the protocol is safe. Thus, using the AVISPA tool with OFMC and CL-AtSe backends, along with a bounded number of sessions, our results show that the proposed protocol is secure from replay and man-in-the-middle attacks.

VII. PERFORMANCE ANALYSIS

Performance of the proposed protocol is analyzed in terms of storage, computational, and communication cost respectively. Since RFID reader and supply chain nodes have high processing power, the performance of resource-constrained RFID tags is thus analyzed.

A. STORAGE COST

This particular cost refers to the cost that an RFID tag incurs by storing data that it needs prior to deployment. In our protocol, an RFID tag needs to store K and IDS each 96 bits long. Therefore, the total storage cost is merely the cost of storing 192 bits, which is considerably smaller compared to protocols described in [37].

B. COMPUTATIONAL COST

Let T_{xor}, T_{hw}, and T_{rot} denote the time needed for executing an exclusive-OR, hamming weight and rotation operations respectively. During the mutual authentication process within the authentication phase, an RFID tag has a computational cost of $6T_{xor} + 9T_{hw} + 7T_{rot}$ and $5T_{xor} + 6T_{hw} + 4T_{rot}$ for the data update process. Therefore, the total computation cost, T_{comp} of an RFID tag during the authentication phase is given by $11T_{xor} + 15T_{hw} + 11T_{rot}$. Lee *et al.* [38] deduced that the computational cost of an exclusive-OR operation, T_{xor} can be ignored because this cost is substantially less than that of using one-way hash functions, T_h and symmetric encryption, T_{enc}. Furthermore, since both hamming weight and rotation operations are bitwise operations, T_{hw} and T_{rot} are themselves negligible and thus the computational cost T_{comp} of the proposed protocol is considered to be negligible.

C. COMMUNICATION COST

Communication is initiated with a 40 bit Hello message and a 96 bit random number T, sent from a reader to the tag.

TABLE 4. State of the art and proposed protocol comparison.

Description	Proposed protocol	Tewari and Gupta [21]	Lin et al. [23]	Hsu et al. [25]	Mujahid et al. [26]	Mujahid et al. [2]
Security protection from						
Key disclosure	Yes	No	Yes	Yes	Yes	Yes
Replay	Yes	Yes	Yes	No	Yes	Yes
MITM	Yes	No	Yes	No	Yes	Yes
De-Sync	Yes	No	No	No	No	Yes
Tracking	Yes	No	Yes	Yes	Yes	Yes
Performance						
Storage cost (bit)	192	384	384	424	672	672
Computational cost (seconds)	$11T_{xor} + 15T_{hw} + 11T_{rot}$	$12\,T_{xor} + 7\,T_{rot}$	$9\,T_{xor} + 4\,T_h$	$4\,T_{xor} + 3\,T_{enc}$	$29\,T_{xor} + 1\,T_{hw} + 29\,T_{rot}$	$27\,T_{xor} + 1\,T_{hw} + 25\,T_{rot}$
Communication cost (seconds)	520	616	576	512	520	520
Blockchain-enabled	Yes	No	No	No	No	No

The proposed protocol performs mutual authentication using four messages (A, B, C, and D) each 96 bits long. Therefore, the total communication cost is merely the cost of transmitting 520 bits, and is significantly lower compared to the protocols mentioned in [37].

The proposed protocol was compared with existing lightweight authentication RFID protocols in terms of security, features and performance, as shown in Table 4. As can be seen, both the proposed protocol and the KMAP+ one [2] are the only two that are able to protect RFID systems from all 5 security attacks. However, the proposed protocol outperforms KMAP+ as our approach requires the lowest storage cost among all existing state of the art protocols and is the only protocol that was designed to be integrated into the blockchain.

VIII. CONCLUSION

This paper presented a secure ultra-lightweight RFID protocol targeted for integration in a supply chain management system that utilizes permissioned blockchain network. Unlike traditional centralized databases, the secret data is encrypted before being written to the permissioned blockchain. Supply chain nodes are split into four categories, namely manufacturer, distributor, retailer, and end user. Each supply chain node is granted with different access levels. Hence, to differentiate easily between them, a random number with either odd or even hamming weight is used to represent access levels of the nodes.

The proposed protocol has been proven to be robust as the probability of data collision stored in the blockchain in the worst case scenario is close to negligible. In addition, the proposed protocol has been proven to be immune to five attacks using both general and formal analyses. The attacks include key disclosure, replay, man-in-the-middle, de-synchronization, and tracking. The proposed protocol has been proven to be efficient in terms of storage, computational, and communication costs. Therefore, it is deemed to be suitable for implementation in supply chains with permissioned blockchain networks.

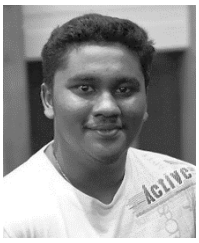
REFERENCES

[1] W. T. Chen, "A feasible and easy-to-implement anticollision algorithm for the EPCglobal UHF class-1 generation-2 RFID protocol," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 485–491, Apr. 2014.
[2] U. Mujahid, M. Najam-ul-Islam, and M. Khalid, "Efficient hardware implementation of KMAP+: An ultralightweight mutual authentication protocol," *J. Circuits, Syst. Comput.*, vol. 27, no. 2, p. 1850033, 2017.
[3] F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, "Privacy-preserving mutual authentication in RFID with designated readers," *Wireless Pers. Commun.*, vol. 96, no. .3, pp. 4819–4845, Oct. 2017.
[4] J. Khor, W. Ismail, and M. G. Rahman, "Detecting counterfeit RFID tags using digital forensic," in *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications*, vol. 555. New York, NY, USA: Springer, 2014, pp. 211–251.
[5] J. H. Khor, W. Ismail, and M. G. Rahman, "Prevention and detection methods for enhancing security in an RFID system," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 8, p. 891584, 2012.
[6] J. H. Khor, W. Ismail, M. I. Younis, M. K. Sulaiman, and M. G. Rahman, "Security problems in an RFID system," *Wireless Pers. Commun.*, vol. 59, no. 1, pp. 17–26, Jul. 2011.
[7] N. Kumar, K. Kaur, S. C. Misra, and R. Iqbal, "An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud," *Peer-Peer Netw. Appl.*, vol. 9, no. 5, pp. 824–840, Sep. 2016.
[8] I. Erguler and E. Anarim, "Security flaws in a recent RFID delegation protocol," *Pers. Ubiquitous Comput.*, vol. 16, no. 3, pp. 337–349, Mar. 2012.
[9] C. K. H. Lee, K. L. Choy, K. M. Y. Law, and G. T. S. Ho, "Application of intelligent data management in resource allocation for effective operation of manufacturing systems," *J. Manuf. Syst.*, vol. 33, no. 3, pp. 412–422, Jul. 2014.
[10] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017.
[11] N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Prof.*, vol. 19, no. 4, pp. 68–72, 2017.
[12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://satoshinakamoto.me/2008/11/01/bitcoin-p2p-e-cash-paper/ and http://www.bitcoin.org/bitcoin.pdf
[13] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[14] J. H. Park and J. H. Park, "Blockchain security in cloud computing: Use cases, challenges, and solutions," *Symmetry*, vol. 9, no. 8, p. 164, 2017.
[15] N. Herbaut and N. Negru, "A model for collaborative blockchain-based video delivery relying on advanced network services chains," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 70–76, Sep. 2017.
[16] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2017, pp. 464–467.
[17] E. Karafiloski and A. Mishev, "Blockchain solutions for big data challenges: A literature review," in *Proc. 17th Int. Conf. Smart Technol.*, Ohrid, Macedonia, Jul. 2017, pp. 763–768.
[18] W. Xin, T. Zhang, C. Hu, C. Tang, C. Liu, and Z. Chen, "On scaling and accelerating decentralized private blockchains," in *Proc. 3rd Int. Conf. Big Data Secur. Cloud*, Beijing, China, May 2017, pp. 267–271.
[19] J. Bell, T. D. LaToza, F. Baldmitsi, and A. Stavrou, "Advancing open science with version control and blockchains," in *Proc. IEEE/ACM 12th Int. Workshop Softw. Eng. Sci. (SE4Science)*, May 2017, pp. 13–14.
[20] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.
[21] A. Tewari and B. B. Gupta, "Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," *J. Supercomput.*, vol. 73, no. 3, pp. 1085–1102, Mar. 2017.
[22] J. H. Khor and M. Sidorov, "Weakness of ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," in *Proc. 8th Int. Conf. Inf. Sci. Technol. (ICIST)*, Jun./Jul. 2018, pp. 91–97.

[23] I.-C. Lin, H.-H. Hsu, and C.-Y. Cheng, "A cloud-based authentication protocol for RFID supply chain systems," *J. Netw. Syst. Manage.*, vol. 23, no. 4, pp. 978–997, 2015.
[24] J. H. Khor and M. Sidorov, "Security flaws and improvement of a cloud-based authentication protocol for RFID supply chain systems," in *Proc. 3rd Int. Conf. Comput. Commun. Syst. (ICCCS)*, 2018, pp. 487–491.
[25] C.-H. Hsu, S. Wang, D. Zhang, H.-C. Chu, and N. Lu, "Efficient identity authentication and encryption technique for high throughput RFID system," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2581–2591, 2016.
[26] U. Mujahid, M. Najam-ul-Islam, and S. Sarwar, "A new ultralightweight RFID authentication protocol for passive low cost tags: KMAP," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 725–744, Jun. 2017.
[27] M. Safkhani and N. Bagheri, "Generalized desynchronization attack on UMAP: Application to RCIA, KMAP, SLAP and SASI+ protocols," *Cryptol. ePrint Arch.*, vol. 2016, p. 905, 2016.
[28] A. Juels, R. Pappu, and B. Parno, "Unidirectional key distribution across time and space with applications to RFID security," presented at the 17th Conf. Secur. Symp., San Jose, CA, USA, 2008.
[29] S. Cai, T. Li, C. Ma, Y. Li, and R. H. Deng, "Enabling secure secret updating for unidirectional key distribution in RFID-enabled supply chains," in *Proc. Inf. Commun. Secur.*, Berlin, Germany, 2009, pp. 150–164.
[30] T. Li, Y. Li, and G. Wang, "Secure and practical key distribution for RFID-enabled supply chains," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.*, Berlin, Germany, 2012, pp. 356–372.
[31] K. Toyoda and I. Sasase, "Secret sharing based unidirectional key distribution with dummy tags in Gen2v2 RFID-enabled supply chains," in *Proc. IEEE Int. Conf. RFID (RFID)*, Apr. 2015, pp. 63–69.
[32] D. Buckley and D. Mullen, "EPC radio-frequency identity protocols generation-2 UHF RFID standard," GS1, Brussels, Belgium, Tech. Rep. 2.1, 2018.
[33] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, May 1990, pp. 234–248.
[34] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.
[35] V. Odelu, A. K. Das, K.-K. R. Choo, N. Kumar, and Y. Park, "Efficient and secure time-key based single sign-on authentication for mobile devices," *IEEE Access*, vol. 5, pp. 27707–27721, 2017.
[36] M. Wazid, A. K. Das, M. K. Khan, A. A. D. Al-Ghaiheb, N. Kumar, and A. V. Vasilakos, "Secure authentication scheme for medicine anti-counterfeiting system in IoT environment," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1634–1646, Oct. 2017.
[37] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2831–2843, Nov. 2018.
[38] C.-C. Lee, C.-T. Chen, P.-H. Wu, and T.-Y. Chen, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Comput. Digit. Techn.*, vol. 7, no. 1, pp. 48–56, Jan. 2013.

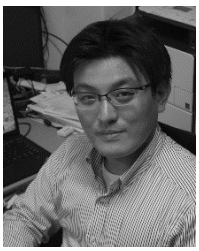
MICHAIL SIDOROV received the B.Sc. degree in informatics from Coventry University, U.K., in 2009, the B.Sc. degree in informatics engineering from Klaipeda University, Lithuania, in 2010, and the joint M.Sc. degree in embedded computing systems from the Norwegian University of Science and Technology, Norway, and the University of Southampton, U.K., in 2013. He is currently pursuing the Ph.D. degree in the IoT with the Toyohashi University of Technology, Japan. He was a Teaching Fellow and a Laboratory Officer with the University of Southampton Malaysia, from 2013 to 2017. His current research interests include energy harvesting sensor networks, LPWAN, and blockchain technology for the IoT.

MING TZE ONG received the B.Sc. degree in math and computing from the National University of Singapore and the M.Sc. degree in numerical analysis from the University of Manchester. He is currently a Mathematics Lecturer with the University of Southampton at Malaysia. His current interests include high-performance computing with parallel systems and the detection of housing market bubbles in Malaysia.



RAVIVARMA VIKNESWAREN SRIDHARAN is currently pursuing the master's degree in electrical and electronics with the University of Southampton. His interests include robotics, control theory, automation, and blockchain technology, as well as cryptocurrency from the economic perspective.

JUNYA NAKAMURA received the B.E. and M.E. degrees from the Toyohashi University of Technology, Japan, in 2006 and 2008, respectively, and the Ph.D. degree in information science from Osaka University, in 2014. He is currently an Assistant Professor with the Information and Media Center, Toyohashi University of Technology. His research interests include distributed algorithms and systems, such as Byzantine consensus and Byzantine fault tolerance.



REN OHMURA received the Bachelor of Electric Engineering, Master of Computer Science, and Ph.D. degrees from Keio University, in 1999, 2001, and 2004, respectively. He is currently a Lecturer with the Department of Computer Science and Engineering, Toyohashi University of Technology (TUT), Japan. He currently runs the Ubiquitous Systems Laboratory, TUT, and supervises students ranging from B.Sc. to Ph.D. level. His research interests include ubiquitous computing, wearable computing, the IoT, smart environment, battery-less systems, and distributed systems.

JING HUEY KHOR received the B.Eng. degree (Hons.) in electrical engineering (electronic) from Universiti Malaysia Pahang, in 2009, and the Ph.D. degree for conducting research on passive radio frequency identification (RFID) security from Universiti Sains Malaysia, in 2013. She is currently an Assistant Professor with the University of Southampton at Malaysia. Her research interests include information security in RFID and the IoT, as well as blockchain technology.