IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 24, 2020, accepted March 14, 2020, date of publication March 31, 2020, date of current version April 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984675

Cyber-Physical Trust Systems Driven by Blockchain

ALEXANDER J. M. MILNE, ARNOLD BECKMANN, AND PARDEEP KUMAR, (Member, IEEE)
Department of Computer Science, Swansea University, Swansea SA1 8EN, U.K.

Corresponding author: Arnold Beckmann (a.beckmann@swansea.ac.uk)

This work was supported in part by the Welsh Government's European Social Fund (ESF) Convergence Programme for West Wales and the Valleys through the Knowledge Economy Skills Scholarships KESS II initiative led by Bangor University.

ABSTRACT Cyber Physical Trust Systems (CPTS) are Cyber Physical Systems and Internet of Things enriched with trust as an explicit, measurable, testable and verifiable system component. In this paper, we propose to use blockchain, a distributed ledger technology, as the trust enabling system component for CPTS. We propose two schemes for CPTSs driven by blockchain in relation to two typical network model cases. We show that our proposed approach achieves the security properties, such as device identification, authentication, integrity, and non-repudiation, and provides protection against popular attacks, such as replay and spoofing. We provide formal proofs of those properties using the Tamarin Prover tool. We describe results of a proof-of-concept which implements a CPTS driven by blockchain for physical asset management and present a performance analysis of our implementation. We identify use cases in which CPTSs driven by blockchain find applications.

INDEX TERMS Cyber physical systems, Internet of Things, distributed ledger technology, blockchain, asset management, supply chain, Industry 4.0, deep leasing.

I. INTRODUCTION

Cyber-physical systems (CPS) are envisioned as an emerging paradigm that focuses on seamless integration and orchestration of objects and embedded systems, communicating with one another using advanced networking technologies [1], [2]. In recent trends, the distinction between CPS and Internet of Things (IoT) is blurred, with CPS serving as IoT devices and IoT devices being components of CPS or vice-versa. A CPS device is a low-cost device (e.g., sensor and actuator, RFID, etc.) integrated with limited computation capabilities, small memory and low bandwidth. CPS devices are being used in various industrial domains, e.g., manufacturing, healthcare monitoring using sensors, medical applications, IoT assisted living, power generation and distribution, smart aircraft, water management systems, asset management, and so on [3]. In distributed applications (as shown in Fig. 1), a significant number of CPS devices will deploy and generate massive volumes of data streams at high speed. These data streams will potentially communicate over the public network (e.g., 4G/5G) to provide a wide range of services, such as reliable data transmission, to their respective applications.

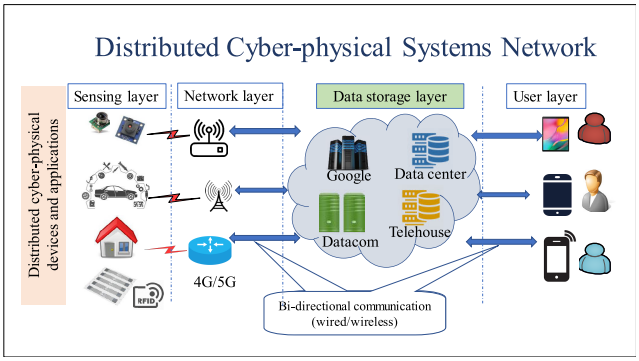

FIGURE 1. Distributed cyber-physical systems.

A. PROBLEM STATEMENT

The deployment of heterogeneous CPS devices promises reliable data transmission, scalability and application efficiency; however, they bring a plethora of security and trust issues for data-driven applications. As shown in Fig. 1, the data

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

generated at the sensing layer is aggregated via different public networking technologies, and owned by several device manufactures (including untrusted) in their data centers. Finally the data is being used by the users. Such CPS networks bring a strong limitation of trustworthiness of data as anonymous attackers can counterfeit fake devices to compromise the network. Recent research shows that attacks (using fake devices) made significant damages to physical processes and brought enormous losses (e.g., performance degradation, etc.) to the CPS network and so to related property [4]. In a similar vein, the authors in [5] implemented a new packet manipulation approach through a dishonest node to inject malicious data to a ZigBee based CPS network. Such ZigBee-based dishonest nodes can even cause a network and/or process control system shutdown, by sending bogus data or commands to a CPS in an industrial domain. In addition, in many CPS applications, the identity of devices forms an important part of the overall ecosystem they are integrated in [6]. Often there are a number of actors, which may be devices or humans, that participate in such ecosystems, and who in general do not trust each other. While some actors may interact with devices directly, they often rely on virtual representations of the device's identities and their data. These virtual identities can pose new concerns in distributed CPS use-cases. For instance, untrusted users or devices may provide nonconforming identities or may maliciously be denied their acquirement of data from the CPS network. Nevertheless, if trust is not being established regarding the identity and overall integrity of the actors, then untrusted data may enter the system and open several risks. For instance, as reported in a media report [7], telemetry vulnerabilities can allow data tampering and interception of other parameters (e.g., identity). Such vulnerabilities may not allow an application to accomplish the desired goal. The challenge in such a situation is how actors can gain trust in the integrity of identities and data in an *explicit, measurable and testable way.*

B. RELATED WORK

To solve the aforementioned security challenges, several methodologies and protocols for enhancing security in the cyber domain have been recently proposed, well-researched, and applied to CPS. These solutions can be categorized as traditional approaches (including trusted third party (TTP) and non-trusted party based solutions) [8]–[10] and blockchain based solutions [11]–[15]. Each approach has its own advantages and disadvantages.

1) TRADITIONAL APPROACHES

Genge *et al.* proposed a security-driven control application in industrial CPS [8]. The main objective is to design a lightweight key distribution scheme that achieves data and device authentication. To do this, the CPS network is divided into groups and each group is controlled and managed by a leader node. Each group leader uses a master key to negotiate a secret key with the other nodes in a group. Unfortunately, the shared master key can easily experience severe risks if a CPS node is compromised, leading to several attacks in the group.

Renuka *et al.* proposed three security mechanisms for machine-to-machine network in CPS [9]. These mechanisms include: (i) mobile-gateway authentication, (ii) mobile-sensor authentication and (iii) sensor-sensor authentication. In the mobile-gateway approach, the authors included a human-in-loop approach where the user is being authenticated using a password. However, in [16], Lara *et al.* claimed that the approaches proposed in [9] require high computational complexities and are vulnerable to several attacks, e.g., off-line guessing attack, privileged insider, and denial-of-service (DoS). In addition, Lara *et al.* proposed another solution to mitigate the issues in [9].

In [10], Wang *et al.* introduced a new concept of optimistic fair exchange (OFE) in CPS. Basically, the main focus of the scheme is online data exchange in cloud-assisted CPS networks. A verifiably encrypted identity-based signature (VEIS) is being used for entity verification. The proposed VEIS uses a centralized TTP that ensures fairness of message exchanges. However, the scheme proposed in [10] may not work without a TTP, therefore, such a scheme may not be practical in real-world distributed CPS applications. Moreover, it is widely known that a TTP may experience performance issues in a large network and may increase risk as it could be a single-point of failure.

Nevertheless, most traditional security mechanisms are either centralized or inefficient for real-world distributed CPS applications, as a centralized security system could be a target for an attacker. Undoubtedly, the above-mentioned traditional security solutions in CPS networks are typically implemented and deployed by third parties or brokers which can impose many security-related risks if the third party is compromised.

2) BLOCKCHAIN APPROACHES

Recently, blockchain as a security-service has attracted more and more attention from both academia and industry [11]–[15]. It is spanning across several domains, including supply chain systems, banking, healthcare, asset management, etc. We present the state-of-the-art work on blockchain based security services (such as authentication, trust, integrity, etc.) in Internet of Things, wireless sensor networks, etc.

Blockchain technologies have transaction-recording and non-duplicability services and thus are a good technological choice for several applications. More precisely, these services demonstrate the suitability of blockchains technologies for public key infrastructure (PKI). Blockchain-based PKI solutions are distributed and have no centralized point of failure. As a result, certificate-based PKI solutions can be used to realize authentication using blockchain [11], [12].

However, public-key certificates have their own shortcomings and issues. In order to solve certificate issues, Lin *et al.* proposed a new solution using blockchain [17]. An identity-based linearly homomorphic signature scheme is designed and implemented to secure the entire network. In this scheme,

a node ID is utilized as a public key of the node. The encryption approach of their proposed scheme consists of four phases: setup, extract, encrypt, and decrypt. All the operations are compute intensive operations. In addition, the scheme is proven to be a safe-guard against existential forgery on adaptively chosen message and identity attack under the random oracle model.

Lewison-Corella proposed a blockchain based distributed database to store data securely [13]. The main idea of the paper is to allow a certificate authority (CA) to publish an unsigned certificate. The blockchain stores the hash value of a certificate and that stored value is controlled by entities, such as banks or governments. These entities make use of two blockchains; typically, one blockchain is used to issue the certificates, and another is used to revoke the certificates. During the certification verification process, an entity first assures the corresponding data is stored to the blockchain. If the certificate's hash value is found in the database, then the certificate is a valid certificate. Otherwise it is not a valid certificate and will be revoked from the blockchain. The authors claimed that the proposed idea is straight forward and it can offer several advantages, such as efficient verification of a certificate with a guaranteed low delay. However, implementation and evaluation results are missing, therefore the viability of this approach is a big question.

Lin *et al.* proposed a blockchain based secure mutual authentication and access control system for Industry 4.0 [14]. They claimed to provide various security services, including anonymous authentication, auditability, and confidentiality and privacy. The authors utilized attribute based signatures to achieve anonymous authentication and fine-grained access control. Lin *et al.* adopted a consensus procedure, which is based on the practical byzantine fault tolerance (PBFT) approach. However, it is widely accepted that PBFT suffers from scalability issues as discussed in [18].

As the number of Internet of Things (IoT) devices explodes, designing a robust and efficient centralized authentication system is almost impossible. Hammi *et al.* proposed a decentralized blockchain-based authentication system for IoT [19]. To achieve their goals, the proposed scheme utilized the security features provided by blockchain, and designed several secure virtual zones (called bubbles). In such zones the smart objects or things can identify each other, establish trust, and protect the system against replay attacks using time-stamps. However, a time-stamp based system may be vulnerable to time synchronization attacks that can lead to further security threats, e.g., a DoS attack.

Another piece of research focuses on blockchain based digital identity management also known as ''BIDaaS: Blockchain based ID as a Service'' [15]. This research mainly targets identity management in mobile telecommunication networks. BIDaaS consists of three different entities: users, the BIDaaS provider, and partners. Here, the user is a mobile user, the telecommunication company is a BIDaaS provider, and the partner is a stakeholder of the telecommunication company. The basic idea of the scheme is that mutual authentication is performed between the user and the partner. Note that the scheme did not have any pre-shared information among entities, therefore it is hard to understand how these entities would verify each other. Moreover, in this scheme, the blockchain server (i.e., BIDaaS provider) utilizes its own public and private key pair to provide the security services (e.g., authentication). However, we have not seen evidence that a blockchain implementation like this is possible without either, the signing being done separately from the blockchain and instead on the server the blockchain runs on (in this case other nodes cannot also verify and trust this action), or a 3rd party being used for performing the private key functionality. The reason signing must not be performed on the blockchain is because the private key would need to be distributed to all nodes in the blockchain for each node to decrypt messages and verify transactions for consensus.

As many IIoT applications consist of resource-constrained devices, network availability and security must be considered. Applying traditional blockchain-based security approaches may pose a challenge to resource-constrained devices. To mitigate this issue, Seok *et al.* [20] investigated a lightweight hash-based blockchain architecture in IIoT. The proposed architecture consists of three layers: (i) field layer (includes sensors and actuators), (ii) control layer (for controlling the devices), and (iii) blockchain layer. Further, the blockchain layer consists of two parts: cell node and storage node. The authors proposed to use a pre-shared signature and several hashing algorithms (e.g., Quark, Photon, and Spongent) at the blockchain layer. However, the proposed scheme involved many hashes which required more computing power. Resultant, it may lead to inefficiency and a significant overhead at the blockchain layer.

In many of the proposed mechanisms [13], [17], there are gaps of viability, unclear principles of blockchain technology and other issues, e.g., performance efficiency issues of security mechanisms (as discussed in related work). Other schemes, e.g., [19], make use of time-stamp, which may be vulnerable to time synchronization attacks. Note that none of the aforementioned blockchain-based approaches are provably verified or implemented. Therefore, there is a need to design more appropriate security solutions that provide explicit, measurable, testable and verifiable trust.

C. MOTIVATION AND CONTRIBUTION

Motivated by the aforementioned challenges, our aim is to establish CPS solutions that provide explicit, measurable, testable and verifiable trust. Following Beckmann et. al. [21], a CPS that has explicit built-in mechanisms for providing trust in the integrity of identities and data, is called a *Cyber Physical Trust System (CPTS)*. *Trust* can be defined as *reliance on the character, ability, strength, or truth of someone or something; one in which confidence is placed* [22] or as *the firm belief in the reliability, truth, or ability of someone or something* [23]. In the context of CPS, we interpret trust to mean *the firm belief in the reliability and truth of data produced by those CPS devices*. Based on that

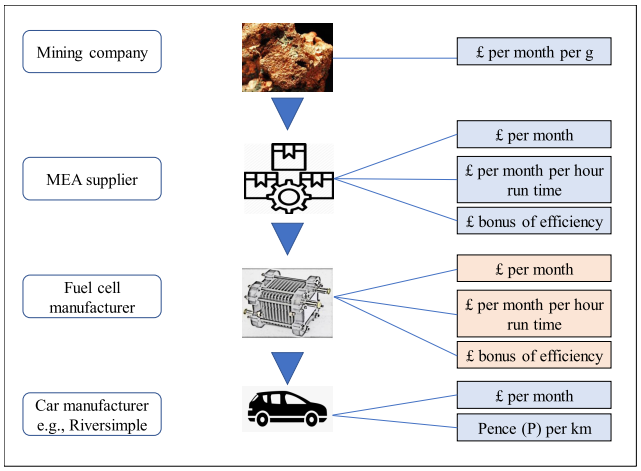

FIGURE 2. Vision of replacing ownership of assets with a deep leasing supply chain in which components are provided as services, as proposed by Riversimple for its hydrogen fuel cell powered Rasa.

interpretation of trust, and a definition of CPS given in [1], CPTS has been defined as follows:

Definition 1 (A Cyber Physical Trust System CPTS [21]): A Cyber Physical Trust System *integrates computation, networking, physical processes, and explicit mechanisms for gaining trust in integrity of data about processes.*

CPTSs need to be contrasted with other notions like the trustworthiness of CPSs, which is the combination of security, privacy, safety, reliability, and resilience [24]. Trustworthiness is a property which is implicit to a CPS, often established as a form of certificate. It usually cannot be tested on a CPS system level but exists externally to it.

In the process of defining CPTS, we have co-created use cases in collaboration with our industrial partners, Oyster Bay Systems [25] and Riversimple [26], that show the importance and relevance of CPTS. In fact, the definition of CPTS has been developed in this co-creation process. Oyster Bay Systems is a fin tech company that specializes in vehicle leasing products. As shown in Fig. 2, Riversimple is a company in Wales developing a vehicle prototype, the Rasa, built with a different manufacturing philosophy: Aiming towards a circular economy, goods like vehicles will be given to customers as a service instead of transferring ownership. That model will also apply to sub-parts of goods: The fuel cell within the Rasa is given as a service while ownership is retained by the fuel cell manufacturer. The membrane (MEA, Membrane Electrode Assembly) in a fuel cell is given as a service to the fuel cell manufacturer while ownership is retained by the MEA manufacturer. And similar for the platinum on the membrane. We call such a service supply chain *deep leasing*. CPTS would be at the heart of making deep leasing work, as trust in the reliability of usage data is a fundamental requirement.

The main contributions of this paper are as follows.

- We describe relevant use cases around deep leasing supply chains that have been co-created in collaboration with our industrial partners.
- We propose two schemes for CPTSs driven by blockchain in relation to two typical network model cases. In the first case, a CPS device is periodically reporting its own data to the blockchain without actor facilitation, while maintaining the identity and data integrity of this CPS device. In the second case, a CPS device is reporting its data on-demand to the blockchain with actor facilitation.
- We provide in depth formal (using Tamarin prover) and informal security analysis of the proposed schemes, which establish that our schemes have the security properties, e.g., decentralization, transparency, unpredictability, device authentication and integrity, device identification, and non-repudiation. In addition, the proposed schemes provide protection against popular attacks. To the best of our knowledge this is the first time that formal proofs verifying security properties of blockchain schemes have been reported.
- We are describing pseudo random values (PRVs) and their relation to nonces, as a contribution to general blockchain research. We describe how they can be realized within typical blockchain systems and what general properties they have. We highlight their usefulness for certifying data freshness, limitations in terms of real randomness, and implications for the analysis of security properties.
- We implement a proof-of-concept to conduct in depth performance analysis using the Dev-System, Surface-Go, and Raspberry-pi. The results show that the proposed schemes are efficient in terms of computational complexities for a resource-constrained device. Finally, we provide a short discussion on real-world applications, which are being used as use-cases for the proposed schemes.

The rest of the paper is organized as follows: Section II presents our underlying network and thread models, and security design goals. Section III gives the necessary blockchain background for the discussions in this paper. Section IV describes our proposed schemes, whose security is then analyzed in Section V. Section VI describes a proof-of-concept and discusses its performance. Section VII presents two use cases in which blockchain based CPTS can be applied. Finally, conclusions are drawn in Section VIII.

II. NETWORK MODEL, THREAT MODEL, AND DESIGN GOALS

The basic design idea of a blockchain based CPTS is that CPS devices are linked to a blockchain ledger which is distributed amongst the actors of the ecosystem. The key blockchain features will ensure that data stored on the blockchain and smart contracts executed by the blockchain are trusted amongst the actors.

A. NETWORK MODEL

Fig. 3 gives an overview of the network model for the proposed schemes. We distinguish two typical cases in which

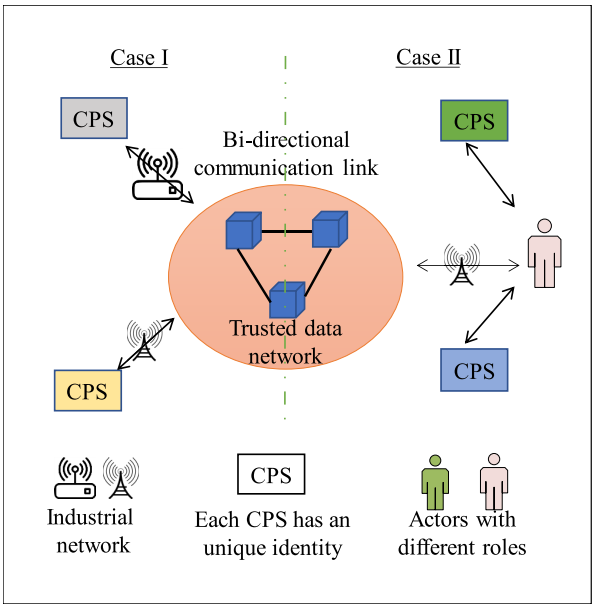

FIGURE 3. The network model for connecting a blockchain based trusted data network with CPS devices, with (in Case II) and without (in Case I) actor facilitation.

CPS devices can interact with a trusted data network. In the first case this is happening directly via an industrial communication network; the second case considers the situation where the communication is happening via actors. For both cases, the network model includes the following entities:

- *Trusted data network:* A blockchain system provides a trusted data network storing relevant information generated by the network. This information will include the identities of actors and CPS devices, as well as data produced by CPS devices.
- *Actors:* Actors are users interested in the CPTS. Actors can be hand-held devices (like mobile phones) that directly interact with CPS devices. They can also be remote clients retrieving information from the trusted data network without direct access to CPS devices. All actors have computing capabilities, e.g. to perform basic cryptographic operations. They have cryptographic identities which are registered on the blockchain and are linked to the trusted data network.
- *CPS devices:* A CPS device is an intelligent device that is integrated with a sensing unit, a computational unit, and communication capabilities. We assume that all CPS devices have cryptographic identities which are registered on the blockchain, and that they are able to communicate their identities in form of a public key, to communicate data related to their processes, to receive additional data, and to sign data (e.g. their process data or received data). The CPS device periodically records the generated/sensed data to the blockchain network, either using the industrial network (Case I), or by interaction with actors (Case II).
- *Industrial network:* In this network model, the CPS devices communicate to the blockchain network through the industrial network. The industrial network may utilize the following interfaces: (i) IEEE 802.15.4 interface – it is a short-range wireless interface that maintains the communication link for the (CPS) devices. (ii) Wi-Fi or GPRS interface – it is a long-range communication interface, which maintains a communication link for the blockchain network. Furthermore, there may be direct NFC communication between CPS and handheld devices.

B. THREAT MODEL

In a CPS network, assume an adversary (i.e., Dolev-Yao attack model) or a malicious entity who has total control over the network. They can selectively eavesdrop on communications and send fake transactions to the blockchain network. The attacker can replay, inject and modify messages either to the CPS devices side or to the blockchain side. In addition, an attacker can also spoof the identity of CPS objects. By doing this, the attacker might gain unauthorized data access to CPS devices or perform service degradation that may lead to denial-of-service.

C. DESIGN GOALS

Following the aforementioned network model, attack vector and literature survey, a secure system must be designed with the security goals to attain the sustainability and resiliency of the CPS. Therefore, this subsection describes the main design and security goals, as follows.

- *Decentralization of networks*: A CPS network should not rely on a centralized entity anymore [27]. This is due to the fact that centralized environments, where everything is done in the centralized location, may cause performance issues. To achieve performance efficiency and quick decision making, decentralization is highly required in real-world CPS applications.
- *Transparency*: In general, centralized systems are prone to fraud [28]. Therefore, the CPS data must be immutable and transparent.
- *Unpredictability*: In CPS, the entities must be unable to predict each other's transaction in the distributed network, to assert that transactions were sent recently or to assert unlinkability between two entities. Therefore, unpredictability is highly required in distributed CPS where privacy is one of the main concerns.
- *Data authentication*: In real-world cyber physical systems, message authentication is an important goal. Since a malicious user can easily inject fake data to a CPS device, blockchain-based systems must ensure data authenticity and check whether the data has originated from the trusted or claimed source. In general, data authentication allows a receiving entity to check the legitimacy of data and that the data really was sent by the claimed entity.

- *Data integrity*: Data integrity ensures the receiver that the data/transaction received has not been altered by an adversary.
- *Secure identity management*: A massive number of devices will be deployed in a CPTS network, and each CPS device will have its own identity. However, identity management can play a major role in a real-world CPS network to track and trace the information/status of the devices. Therefore, secure identity management is an important requirement for a blockchain-based CPTS.
- *Non-repudiation*: This goal ensures that neither a CPS, blockchain nor an actor can deny any given action that has been performed by them.
- *Data freshness*: Generally, CPS devices transmit data periodically and therefore there must be a mechanism that ensures the data received from a CPS device is recent.
- *Protection against popular attacks*: In real-world environments the CPS should defend against several attacks.

III. BLOCKCHAIN BACKGROUND

Distributed Ledger Technologies are peer to peer networks where multiple independently motivated parties each store some shared data on their own system, but with the guarantee that everyone else is also storing identical data. This will also continue to be true for any newly added data. Blockchains are specific implementations of this. They use connected blocks to form a cryptographically secure chain. There are many different blockchain implementations, the original, Bitcoin [29] and other notable implementations like Ethereum [30], and both Hyperledger Fabric [31] and Sawtooth [32].

A blockchain stores transactions that participants in the network have agreed to be valid and organizes them into immutable blocks. To get meaningful data from these blocks of transactions each party in the network has to build up their own database, usually in the form of Merkle trees [33]. These are built up using the information from each transaction from the first block until the last in the correct order. This ensures that every independent party ends up with identical data. The list of transactions can be seen as an ordered list of instructions, which builds the database from no data to the current state that everyone else holds.

A block consists of a header and a list of transactions. The header contains block information such as a hash of the previous block and other information that differs between different blockchain implementations. The preceding block hash being part of each block links blocks together and creates a chain all the way back to the first block, confirming the integrity of the blockchain. The previous block hash is also what makes the blockchain immutable because any edit to a previous block would change the hash of the block, breaking this mentioned chain.

Consensus is the mechanism that blockchains use for adding new blocks. It dictates what can be done in these blocks, for example only valid transactions are allowed, and decides who is allowed to propose the next block.

The most common type of consensus is proof of work. In this scheme many *miners* will attempt to form a new block, which contains valid transactions, by changing some nonce value in their proposed block and hashing the block. Miners *win the lottery* and get to be the creator of the next block when the produced hash has a hard-enough difficulty, e.g. the hash has 28 zeros at the start of it. The difficulty in the network is changing such that there will be a consistent average block time (10 min for bitcoin [29]). Once they produce a hash with a hard-enough difficulty the block is sent out to the network where everyone will validate the block and add it to their chains. If the miner was too slow, as another miner has produced a block, then they will have to start the process again after adding the new received block. In the case where 2 blocks are created at the same time, some people will mine from the starting point of one of the blocks and some from the other. Everyone in the network agrees that the longest chain is the correct one and so whichever block gets another block added to it faster will become the true blockchain.

Proof of elapsed time (PoET) is the default consensus used for Hyperledger Sawtooth [32] that uses a *trusted execution environment*, Intel SGX. Each node in the network will run code in this secure enclave that will give them a random wait time that can be proved to be fair. The node with the shortest wait time will be the creator of the next block. Each other node upon receiving this block will check that the wait time was run from the SGX and that all other aspects of the block are valid.

Hyperledger Sawtooth [32] is an open source blockchain implementation originally made by Intel and now under the Hyperledger umbrella. Sawtooth can be used for public or private networks and stores the settings that specify permissions, such as roles and identities, so that all participants in the network can access this information. Sawtooth allows for smart contracts that are written as *transaction processors* (TP) that can be written in any language. Unlike other blockchains with smart contracts, such as Ethereum, which create and specify smart contracts using transactions, TPs are programs that must be run on each node in the network and must be identical. More detailed survey papers on security services using blockchain can be found in [27] and [34].

A. COUNTERS AND PSEUDO RANDOM VALUES - NONCE-ESQUE FUNCTIONALITY

A nonce, *number only used once*, has the purpose of ensuring message freshness. This can be split up into two sub-goals, firstly to protect a receiving entity against replay attacks by allowing identification of identical messages, and secondly to ensure that messages have been created and sent recently because nonces expire after some time.

Due to the lack of randomness in blockchain technology [35], providing nonces is a challenge. With a blockchain system, a nonce's first goal can be replaced by a counter that is sent and increased with each message ensuring a unique and ordered number only used once is contained in the message. However, a counter cannot be used to completely replace a

nonce because it does not fulfill a nonce's secondary goal of ensuring messages were created recently. For this we will still need the blockchain to produce some Pseudo Random Value (PRV) that is an unpredictable value that changes frequently. It is almost like a nonce except it can be used multiple times by multiple entities within a time frame. A PRV is used instead of directly using a nonce because it is difficult for blockchain systems to produce random values. The difficulty of producing randomness can be seen with many blockchain systems using a proof of work mining system which is essentially a very expensive random number generator that chooses the producer of the next block. Unpredictability is difficult because for blockchains to have consensus, all involved parties and nodes must agree on the outcome of every element that creates the next block in the blockchain. Thus, every blockchain function must be deterministic making generating random numbers a major issue for blockchain technologies [35]. For an example function, generateRandomNo(), every node must produce the same result deterministically, thus, making it not a random number. However, a blockchain system on the macro, total system, level is not completely predictable despite the fact its code is deterministic. Unpredictability is added into the system by users of the system when they perform actions and transactions. We need a method of harnessing this user-based unpredictability and to use it as a seed for the deterministic code to produce random values. With the idea of harnessing user-based randomness, we argue that the current root hash of the blockchains state, Merkle root hash value or current block hash, cannot be easily predicted. We argue this because there are many transactions, by many users, taking place that are unknown to an attacker. This means that the PRV root hash is changing with each block, making it a suitable replacement for a random assurance of recency from a nonce in typical communication protocol. The PRV will change frequently but not frequently enough to allow for one value for each protocol interaction like a nonce. Thus, we allow the PRV value to be used in multiple communications but expiring after *time*, with it only being valid for a set number of succeeding blocks. With the counter and PRV we replace both the functions of a nonce with the counter solving the replay attack functionality and the PRV solving the recency.

IV. PROPOSED SCHEME

In this section, we introduce blockchain based CPTSs. Consider a CPS network that consists of a number of low-powered CPS devices which sense data from their respective environments and record this data to the blockchain. Following the network model (Fig. 3), we propose two distinct scenarios: (1) A CPS device periodically recording data to the blockchain without actor facilitation; and (2) a CPS device recording data on-demand to the blockchain with actor facilitation. However, before describing our schemes, we first introduce the system registration phase.

Table 1 lists the notations used in the following section and the rest of the paper.

TABLE 1. Notations.

Notations	Descriptions
CPSPk	Public key of CPS device
CPSLtk	Private key of CPS device
TUPk	Public key of Trusted User
TULtk	Private key of Trusted User
Sign	Signature using private key
PRV	Pseudo random value

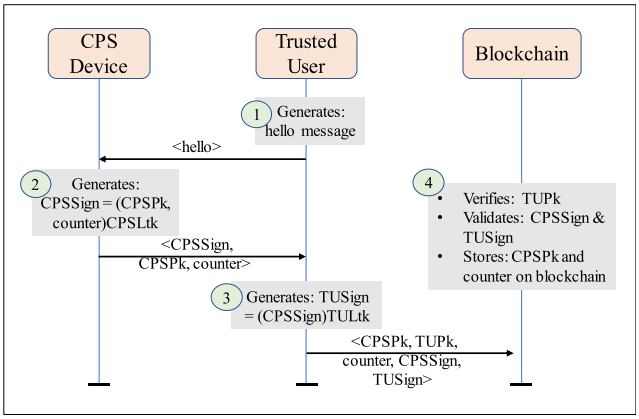

FIGURE 4. CPS device registration to the blockchain.

For our protocols we have assumed the following:

1) Trusted users and CPS devices have credentials (public and private keys).
2) The blockchain has already been set up and is running and secure.
3) Trusted users have been pre-registered on the blockchain.
4) The blockchain has some method of forming PRVs.

A. SYSTEM REGISTRATION PHASE

In this phase, a CPS device is registered on the blockchain by a trusted user with the following steps. The flow of the registration phase is depicted in Fig. 4.

1) A user generates a $\langle hello \rangle$ message and sends it to the CPS device.
2) Upon receiving the message, the CPS device generates a signature, $CPSSign = (CPSPk, counter)CPSLtk$, using its own private key, $CPSLtk$, and sends $\langle CPSSign, CPSPk, counter \rangle$ to the trusted user. Here, $counter$ is a value that is incremented by the CPS device, and used to protect against replay attacks.
3) Now the trusted user generates a signature $TUSign$ on $CPSSign$ using its own private key $TULtk$ and sends $\langle CPSPk, TUPk, counter, CPSSign, TUSign \rangle$ to the blockchain.
4) Upon receiving the message, the blockchain performs the following steps.

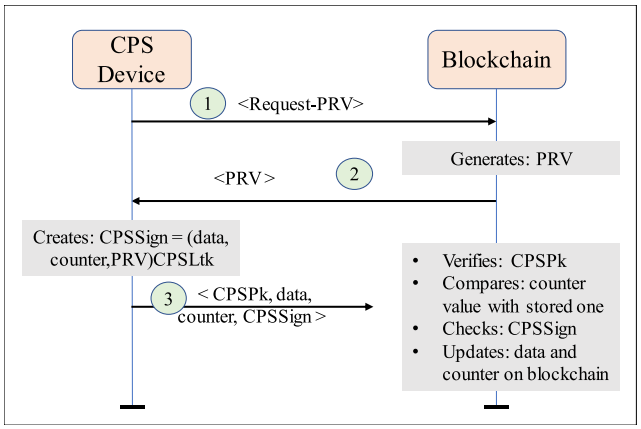

FIGURE 5. Case 1: The CPS device periodically recording data to the blockchain without actor facilitation.

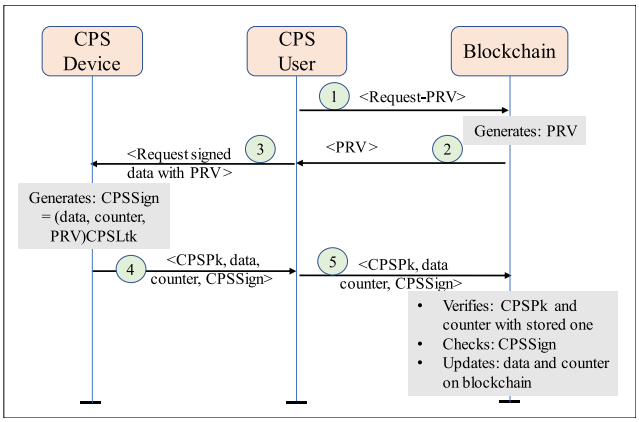

FIGURE 6. Case 2: On-demand recording of CPS device data to blockchain, requested and facilitated by an actor (CPS user).

- First, it verifies the public key of the trusted user *TUPk*. It then checks whether the received public key *CPSPk* of the CPS device has been previously registered on the blockchain. If the CPS device has not been registered, then it validates *CPSSign* and *TUSign*, otherwise aborts the registration phase.
- Finally, if the previous validation has been successful, the blockchain stores *CPSPk* and *counter*.

B. DATA TRANSACTION FROM CPS DEVICES TO BLOCKCHAIN

Case 1 Periodically Recording of CPS Device Data to the Blockchain Without Actor Facilitation

In many industrial automation systems, a CPS device is required to record and update its own data periodically (e.g., every 15/30/45 minutes) to the blockchain. For instance, in modern automobiles a milometer can periodically report its mileage data to the blockchain. In such scenarios, a CPS device records its own data to the blockchain using the following steps. The flow of the scheme is shown in Fig. 5.

1) A set period of time without an update has passed, and thus the CPS device initiates a blockchain data update. It sends a request $\langle Request\text{-}PRV \rangle$ for a pseudo random value to the blockchain.
2) The blockchain, after receiving the request, generates a pseudo random value *PRV* and sends $\langle PRV \rangle$ to the CPS device.
3) Upon receiving $\langle PRV \rangle$, the CPS device generates a signature $CPSSign = (data, counter, PRV)CPSLtk$ using its own private key *CPSLtk*, and sends the message $\langle CPSPk, data, counter, CPSSign \rangle$ to the blockchain.
4) The blockchain receives a message of the form $\langle CPSPk, data, counter, CPSSign \rangle$, stores the data against the CPS device and updates the counter associated with the CPS device, if the following steps are successful:
   - The blockchain first verifies that *CPSPk* is registered in its database. Then it compares *counter* with the counter value stored in its database against *CPSPk*. If *counter* is not greater than the stored counter then the system aborts. Otherwise, it proceeds to the next step.
   - Now the blockchain verifies whether the signature *CPSSign* is signed by the private key *CPSLtk* of the CPS. If true, then the system updates the data and counter against *CPSPk* on the blockchain.

Case 2 On-Demand Recording of CPS Device Data to the Blockchain, Requested and Facilitated by an Actor

In the following we describe a scheme in which data from a CPS device will be recorded on the blockchain. For this scheme we assume that the recording is requested by one of the actors, a CPS user, who has an interest in the data to be documented at this point in time. An example use case for this protocol is for a leasing system where costs are based on some usage data. When the item is leased or returned a CPS user will want to save the data from the CPS device to the blockchain at that exact moment and not wait for a periodic update.

The scheme operates in the following six steps. The flow of the proposed approach is depicted in Fig. 6:

1) The CPS user initiates the communication and sends a pseudo random value request $\langle Request\text{-}PRV \rangle$ to the blockchain.
2) Upon receiving the request the blockchain generates a pseudo random value *PRV* and sends $\langle PRV \rangle$ back to user.
3) The user forwards *PRV* to the CPS device and requests the CPS device to sign its current process data together with the received *PRV*.
4) The CPS device generates *CPSSign* of the form $(data, counter, PRV)CPSLtk$ and sends $\langle CPSPk, data, counter, CPSSign \rangle$ to the user.
5) The user passes this message on to the blockchain.
6) Blockchain executes a smart contract to check authenticity of identity, data and time with the following steps:

- It verifies that *CPSPk* is registered in its database. It then verifies that the received counter value is greater than the stored counter. If both conditions are true proceed to next step, otherwise abort.
- Now the blockchain verifies whether the signature *CPSSign* is signed by the private key *CPSLtk* of the CPS. If true, then the system updates the data and counter against *CPSPk* on the blockchain.

V. SECURITY ANALYSIS

We perform the security analysis of our proposed schemes by combining formal and informal approaches. For the formal analysis, we employ Tamarin Prover, a tool that allows for symbolic modeling and analysis of security protocols. Note that we are intentionally omitting formalizing our design goals Decentralization, Transparency and Unpredictability, as it is widely known that blockchain itself inherits them.

By default, Tamarin has a Dolev-Yao adversary network model, and by using this, can verify or falsify specified properties (lemmas) based on a model (rules). We use Tamarin as it supports the explicit modeling of a state, for example the last counter the blockchain has stored for a specific CPS device. When a counter gets updated, the old value needs to be erased from the model, which Tamarin is supporting, but many other formal verification tools are not.

Tamarin uses rewrite rules on multisets of facts to model protocols, i.e. Input/Output behavior, long-term keys, short-term keys, etc. A fact $F(t_1, \ldots, t_k)$ consists of a fact symbol F of arity k and terms $t_1, \ldots, t_k$. A set of reserved fact symbols is used to denote freshness information (Fr) and messages to the network (In and Out). Other facts are used to represent the protocol state. Linear facts can be consumed only once, persistent facts can be consumed arbitrarily often and are marked with an exclamation mark. Multiset rewriting rules are labeled by so-called actions. The rules consist of premises l, actions a and conclusions r, and are denoted $l-[a]\rightarrow r$. For more information and explanations on Tamarin Prover see [36].

A. FORMAL ANALYSIS USING TAMARIN

For our analysis we assume the following:

1) CPS devices are secure, i.e., they cannot be compromised.
2) The blockchain system is secure.
3) The PRV values are practically fresh at the time of their generation, i.e., they cannot be guessed by any party.

We first explain the rules that model the CPS devices' actions in the Tamarin model. For the counter maintained by a CPS device we make use of Tamarin's multiset feature. We model the counter as a multiset consisting only of the symbol $'1'$, i.e., $'1'$, $('1'+'1')$, $('1'+'1'+'1')$ etc. The cardinality of the multiset is the value of the counter. Thus, one counter is smaller than another if the first is a subset of the latter. We enforce this semantics by adding a restriction that enforces, for all instantiations of rules annotated with Smaller(x, y), that x is a subset of y:

$$\forall x, y, i.\ \text{Smaller}(x, y)@i \Rightarrow \exists z.\ x + z = y$$

Note that i and j range over timepoints. Hence Action@i implies that the trace contains a rule that produces the action Action at timepoint i.

1) CPS DEVICE INITIALIZATION

We describe the protocol that models the initialization of a new CPS device. This does not include the registration phase and thus the blockchain simply registers the device here. In the rule CPS_init, a fresh private key, *CPSLtk*, called long term key, is generated along with its corresponding public key, *CPSPk*. The device permanently saves its public and private key pair in CPSStr and its current (non permanent) counter in CPSCtr. The initial counter value is stored in BCRegCPS on the blockchain – we use an additional fresh variable $\sim l$ to model that creation and consumption of this fact are linked. Finally, the public key of the CPS device is publicly given while the private key stays hidden and never leaves the device.

$$\begin{aligned}&\text{Fr}(\sim CPSLtk),\ \text{Fr}(\sim l)\\ &\quad -[\ \text{CPSReg}(CPSPk)\ ]\rightarrow\\ &\qquad !\text{CPSStr}(CPSPk, \sim CPSLtk),\\ &\qquad \text{CPSCtr}(CPSPk, ('1')),\\ &\qquad \text{BCRegCPS}(CPSPk, '1', \sim l), \text{Out}(\langle CPSPk\rangle)\end{aligned}$$

On each trace this rule can only be instantiated once. This can be enforced using a suitable restriction which we omit here (it is included in our source files [37]).

2) CPS DEVICE SIGNING

The rule CPS_signing models a CPS device signing some stored data $\sim CPSdata$, together with some received data *PRV*, and to send it subsequently to the blockchain. Here the CPS device takes in a value *PRV* and outputs a *message* with a generated signature. Contained in the signature are the CPS device's public key *CPSPk*, the data $\sim CPSdata$, the CPS device's current counter *tc*, and *PRV* from the input. It then saves its incremented counter value after sending the signature with the unsigned information as well to the blockchain for verification.

$$\begin{aligned}&\text{In}(\langle PRV\rangle), \text{In}(tc), \text{Fr}(\sim CPSdata),\\ &\quad \text{CPSCtr}(CPSPk, tc), !\text{CPSStr}(CPSPk, CPSLtk)\\ &\qquad -[\ \text{CPSReceivePRV}(CPSPk, PRV),\\ &\qquad\quad \text{CPSSignData}(CPSPk, tc, sig),\\ &\qquad\quad \text{CPSSendMessage}(CPSPk, message)\\ &\qquad]\rightarrow \text{CPSCtr}(CPSPk, ('1'+tc)), \text{Out}(message)\end{aligned}$$

We now explain the rules that model the blockchain in Tamarin.

3) REQUEST PRV FROM THE BLOCKCHAIN

The rule BC_Init models the blockchain broadcasting its current PRV. In this rule a fresh PRV is initialised, as $\sim PRV$, and is saved by the blockchain in BCPRV. As blocks may be updated only periodically, it is possible that such a requests results in the same PRV being send as the previous one. We model this by allowing BCPRV($\sim PRV$) to be used twice. Finally, $\sim PRV$ is also broadcast to the public network.

$$\begin{aligned}&\text{Fr}(\sim PRV)\\ &\quad -[\ \text{BCPRVOut}(\sim PRV)\ ]\rightarrow\\ &\qquad \text{BCPRV}(\sim PRV), \text{BCPRV}(\sim PRV), \text{Out}(\sim PRV)\end{aligned}$$

4) BLOCKCHAIN AUTHENTICATION OF VALUES SENT BY CPS DEVICE

The final rule BC_Authentication models the blockchain accepting a transaction containing the values send by the CPS device. The blockchain takes in a message. It verifies that the signature has come from a CPS device that has been pre-registered and that the counter received is larger than the one it has saved for this CPS device, to avoid replay attacks. In the rule the message m stands for $\langle CPSPk, CPSdata, tc\_in, sig\rangle$, while $test$ stands for $\langle CPSPk, CPSdata, bc\_PRV, tc\_in\rangle$.

$$\begin{aligned}&\text{In}(m), \text{BCPRV}(bc\_PRV), \text{Fr}(\sim l),\\ &\quad \text{BCRegCPS}(CPSPk, stored\_tc, o)\\ &\qquad -[\ \text{Eq}(\text{verify}(sig, test, CPSPk), \text{true}),\\ &\qquad\quad \text{Smaller}(stored\_tc, tc\_in),\\ &\qquad\quad \text{BCAuthRegValue}(CPSPk, stored\_tc, tc\_in),\\ &\qquad\quad \text{AuthenticSig}(CPSPk, tc\_in, sig),\\ &\qquad\quad \text{AuthenticMessage}(CPSPk, m),\\ &\qquad\quad \text{AuthenticPRV}(CPSPk, bc\_PRV)\\ &\qquad\quad]\rightarrow \quad \text{BCRegCPS}(CPSPk, tc\_in, \sim l)\end{aligned}$$

As explained before, we use Tamarin's restrictions to limit traces to those where counter $stored\_tc$ is smaller than counter tc_in via the action fact Smaller($stored\_tc, tc\_in$) in the above rule. The second restriction we employ is for equality: Eq(s, t) restricts to traces in which term s is equivalent to t modulo Tamarin's underlying equational theory for asymmetric key encryption.

$$\forall x, y, i.\ \text{Eq}(x, y)@i \Rightarrow x = y$$

In the rule above it is used to model that the blockchain will only accept if the signature is valid.

B. INFORMAL AND FORMAL SECURITY ANALYSIS OF THE PROOFS

This section considers the Dolev-Yao attack model while performing formal analyses, providing proofs using the Tamarin model and provided lemmas, and performing informal analyses of the security of the proposed schemes. In this way, we ensure that the design goals mentioned in Section II-C are achieved.

1) AUTHENTIC DATA SOURCE

The authentic data source design goal is, to ensure that all data received by the blockchain system has originated from a trusted, registered, or claimed source. In the proposed schemes, device authentication is performed to keep trust in the system. Following the design goal, the blockchain needs to authenticate the CPS device, otherwise an impersonation attack may occur, which may record fake messages (or tampered message) to the blockchain. In the registration phase, a CPS device sends $CPSSign$ to the trusted user. Then the trusted user appends their own signature $TUSign$ to the CPS device's message, and sends $CPSSign$ and $TUSign$ to the blockchain. This validates the authenticity of the data source since the blockchain checks that both signatures are from registered entities. In Case-1, a CPS device also signs the message, containing its identity, that is passed onto the blockchain by some facilitator. The CPS device also signs the message in Case-2 but this is sent directly to the blockchain. In the above three scenarios all messages are signed using private keys of the sending party, either the CPS device or the trusted user. These private keys are only possessed by the legitimate entities, CPS devices and trusted users, and the blockchain will only accept messages from entities with an identity that it has stored. Therefore, an attacker cannot impersonate either entity.

To formally prove that the data has come from an authentic (pre-registed) source, the Lemma AuthenticDataSource expresses that, if a signature, shown in AuthenticSig($CPSPk, tc, sig$), has been accepted by the blockchain then it must have been signed by a CPS device at some point prior, shown in CPSSignData($CPSPk, tc, sig$), and this CPS device has been registered with the blockchain, shown in CPSReg($CPSPk$), at an earlier point.

$$\begin{aligned}&\forall\ CPSPk, tc, sig, j.\ \text{AuthenticSig}(CPSPk, tc, sig)@j\\ &\quad \Rightarrow\ \exists i\, p.\ (p < i)\ \wedge\ (i < j)\ \wedge\ \text{CPSReg}(CPSPk)@p\\ &\quad \wedge\ \text{CPSSignData}(CPSPk, tc, sig)@i\end{aligned}$$

2) DATA INTEGRITY

The data integrity design goal is to ensure that any data received by the blockchain system has not been tampered with by an adversary when sent over the network. That is, the data sent by a CPS device is identical to the data received by the blockchain. Like the authentic data source goal, this goal is also achieved with the use of signatures. We have already seen in the analysis of the authentic data source goal that the blockchain will only accept messages that contain a signature from an authentic identity. Also, the blockchain checks that all signatures are valid by ensuring that all messages and signatures match up. Thus, the signatures' validity shows that the data has not been tampered with when sent over the network and must have originated from a registered CPS device. Therefore, the data has retained its integrity.

To formally prove that the data is identical when both sent and received, the Lemma DataIntegrity expresses that the

received message (m), shown in AuthenticMessage($CPSPk$, m), is identical to the send message (m), shown in CPSSendMessage($CPSPk$, m), from the CPS device ($CPSPk$). This Lemma is expressed below:

$$\forall CPSPk, m, j.\ \text{AuthenticMessage}(CPSPk, m)@j \Rightarrow \exists i\ .\ (i < j)\ \wedge\ \text{CPSSendMessage}(CPSPk, m)@i$$

3) NON-REPUDIATION

This goal ensures that any action performed by either a CPS device, or an actor cannot be claimed to have been performed by someone other than the effectuator. Non-repudiation is achieved by having all messages signed, the same as with the previous goals above. Since, as mentioned above, all entities keep their private keys hidden and all messages are signed with the sender's private key, only the entity which owns the private key could have signed the message and therefore cannot deny performing the action. Formally this is proven again using the same two lemmas that were previously mentioned above in Paragraphs V-B1 and V-B2. Since the identity is contained in the signed messages that are sent and the signatures are validated using this identity, it shows that the identity in the received message on the blockchain was the identity of the CPS device that signed the message and they cannot claim to have not.

4) SECURE IDENTITY MANAGEMENT

Each entity, device or actor, owns an identity; this is a unique identity. The blockchain stores this identity at the time of its registration utilizing the entity's public keys as its identity. The immutability of information stored on the blockchain guarantees the security of stored identities. The trustworthiness of the identity is assured by the signatures, and it can be seen from Registration, Case-1, and Case-2 that each message sent from an entity contains its public key and is computed over its private key, producing a signature. This private key is only associated to the device identity; therefore, the approach can easily identify entities. Formally, in our model, the identity is stored in the rule CPSInit mentioned in Paragraph V-A1 as a persistent fact. This is then modeled in the BCAuthentication rule where the blockchain will only accept a transaction where the signer is one of the stored identities.

5) RESISTANCE TO SPOOFING ATTACK

Our protocol is also resistant to the common spoofing attack. In this attack, an attacker attempts to spoof another CPS device's identity to accomplish some malicious goals. For instance, in Case 1 (Fig. 5), assume that an ill-intentioned adversary intercepts *message 3*, i.e. $\langle CPSPk, data, counter, CPSSign\rangle$ between the CPS device and the blockchain, and alters it to $\langle CPSPk, dataA, counterA, CPSSign\rangle$. The attacker then attempts to record the altered message containing malicious data to the blockchain. This attempt will not be verified by the blockchain, because $CPSSign$ is computed over the private key of the CPS device. Therefore, spoofing identity does not help an attacker in Case 1, and likewise, spoofing identity does not work in Case 2 either. Formally this is proven in both of the above lemmas for authentic data source, Paragraph V-B1, and for data integrity, Paragraph V-B2. With the assertion that the data has originated from an authentic source and also that the data cannot be tampered with over the network due to the signatures, we can say that the protocol is resistant to spoofing attacks.

6) DATA FRESHNESS AND PROTECTION AGAINST REPLAY ATTACK

Data freshness is the assertion that messages received have been sent recently and that they are not replayed. Normally to accomplish this goal, a nonce is used. But, as mentioned in Section III-A, we cannot use a nonce so instead we split data freshness into its two core components. First, individual messages can be identified and therefore a secondary malicious sending of the same message can be identified and ignored, protecting against replay attacks. The second is to allow some aspect of recency of messages. For example, the message has been sent within a timeframe where the nonce, in our case PRV, is still valid. Typically, a replay attack is the most common threat, where an adversary eavesdrops on communication and copies legitimate messages (e.g., $\langle CPSPk, data, counter, CPSSign\rangle$) recorded between a CPS device and the blockchain. The adversary can replay captured messages at some later time to the blockchain to perform the attack. However, to protect from replay attack, the CPS devices utilize a monotonically increasing counter with each message (e.g., $\langle CPSPk, data, counter, CPSSign\rangle$). Upon receiving a message, the blockchain verifies the received counter value, and if it is an old counter value then the blockchain rejects the message. Thus, an adversary cannot replay old messages in the proposed schemes (Case 1 and Case 2). Therefore, a counter solves the first purpose of a nonce without requiring a random value. For the other aspect of data freshness, and the second purpose of a nonce, we require to know that the message was sent recently. We use a pseudo random value, PRV, generated by the macro blockchain system which uses the randomness of the user's inputs into the system as a seed for a pseudo random number. This cannot be used exclusively instead of a counter, because it can be used by multiple CPS devices in the system, and its generation is independent from the protocol.

For our formal proofs, we firstly prove that for the blockchain to accept a message from a CPS device it requires a correct PRV value, which has been used in signing the message, and that has originated from the blockchain system.

$$\forall\, CPSPk, PRV, v.\ \text{AuthenticPRV}(CPSPk, PRV)@v \Rightarrow \exists\, i, j.\ (i < j < v)\ \wedge\ \text{BCPRVOut}(PRV)@i \wedge\ \text{CPSReceivePRV}(CPSPk, PRV)@j$$

Secondly, we prove that the counters are protecting against replay attacks by showing that a message with the

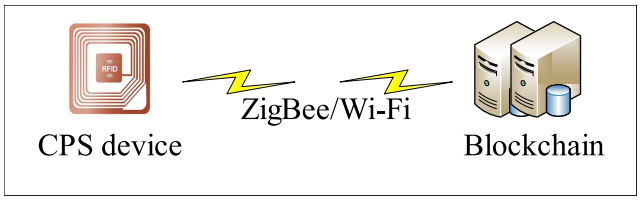

FIGURE 7. Experimental setting using a RFID-based object and the blockchain.

same or smaller counter value will not be accepted by the blockchain system in BCAuthenticate. The following lemma states that when a message has been accepted by the blockchain all smaller counters for a CPS device are invalid, and the system is protected against replay attacks in cases where the counters are smaller.

$$\forall\, CPSPk, oldtc_1, tc_1, oldtc_2, tc_2, i, j, v.$$
$$\text{BCAuthRegValue}(CPSPk, oldtc_1, tc_1)@i$$
$$\wedge\, \text{BCAuthRegValue}(CPSPk, oldtc_2, tc_2)@j$$
$$\wedge\, \text{Smaller}(tc_1, tc_2)@v \;\Rightarrow\; i < j$$

The next lemma expresses the absence of replay attacks where the counter values are identical.

$$\neg \exists\, CPSPk, tc, sig_1, sig_2, i, j.$$
$$i \neq j \;\wedge\; \text{AuthenticSig}(CPSPk, tc, sig_1)@i$$
$$\wedge\, \text{AuthenticSig}(CPSPk, tc, sig_2)@j$$

In order to reach and prove these lemmas about our modeled system we need to prove aspects about the system, such as that the counters are increasing and other properties of the counters and the PRV values. The main lemma has the intuitive meaning that counter values stored on the blockchain are increasing over time. This lemma together with further additional lemmas are provided along with the complete Tamarin source files on our website [37].

VI. PERFORMANCE ANALYSIS

This section discusses the testbed results. First, we discuss the real experimental setting including a CPS device and blockchain. Then performance evaluations are presented, as follows.

A. EXPERIMENTAL SETUP

Our experimental setting is shown in Fig. 7, which consists of an RFID based smart object and the blockchain. In the testbed, a resource-constrained CPS device (i.e., RFID tag plus emulated computational ability) reports its data via either ZigBee or Wi-Fi to the blockchain. For performance comparison purposes, we implemented the system for three different development environments: (i) a desktop windows setup with an i7-6700 @ 3.4GHz and 16 GB of RAM, (ii) a Surface Go "laptop/ tablet" device, integrated with an Intel Pentium Gold 4415Y processor at 1.6GHz and 8 GB of RAM, and

TABLE 2. Table to show computational time.

Node type		Mean Time (ms)	Standard Deviation
Dev System	**Sign**	1.0000	0.0065
	Verify	0.9999	0.0066
Surface Go	**Sign**	0.9994	0.0111
	Verify	1.0560	0.2462
Raspberry Pi	**Sign**	1.5201	0.1977
	Verify	2.0619	0.5929

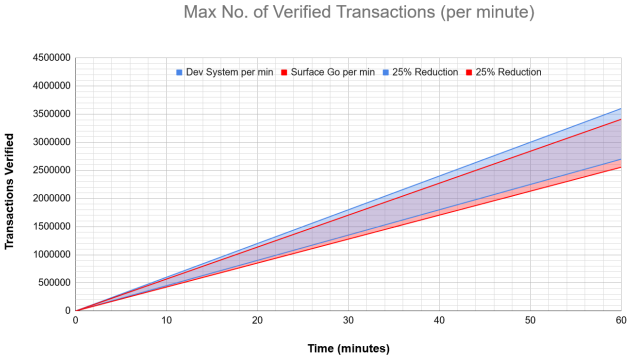

FIGURE 8. The maximum number of transactions that a single machine can verify in minutes.

(iii) a Raspberry Pi, to more closely represent a low powered CPS device. We used a secp256k1 elliptical curve algorithm for signing and verifying the transactions. The Verify functionality will run on the blockchain to verify the signatures in the transactions and the Sign actions are performed on the CPS devices using their data and private keys.

B. COMPUTATIONAL RESULTS

In Table 2, we describe the computation time for signing and verifying. We tested both actions, Verify and Sign, on all three systems although, for example, the Raspberry Pi would not realistically run a blockchain node, therefore it would not in reality run the Verify action. On the development system, the most powerful, the time taken to sign was 1ms with a standard deviation (SD) of 0.0065; this compares to 0.9994ms with 0.0111 SD on the Surface Go and 1.5201ms with 0.1977 SD on the Raspberry Pi.

The Development system performing verify took 0.9999ms with 0.0066 SD, the Surface took 1.0560 with 0.2462 SD and the Raspberry Pi took 2.0619 with 0.5929 SD. We can see that the development system is greatly more powerful but, the very low power in comparison, Raspberry Pi only takes 1.52 times longer to sign data.

Fig. 8 shows the maximum number of transactions that our tested machines can verify in a time period. The graph has an upper and lower bound, with a highlighted area between, for both the Development system and the Surface Go system. The upper bound takes no network delay

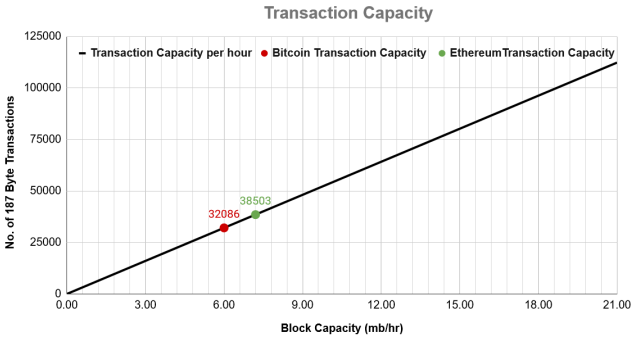

FIGURE 9. Our proposed scheme's maximum theoretical blockchain throughput when implemented on Bitcoin (red) and Ethereum (green).

into consideration. This is because in our implementation there is a direct connection between the CPS device and the blockchain server. Also, in our implementation, there is only one blockchain server, thus there is no delay considered for blockchain overhead. We have made the assumption that there would be a 25% penalty in overhead for both adding more nodes to the blockchain system, and delays related to slower connection between CPS devices and the blockchain. For example, in a 30 minute timespan our Development system is able to verify between 1800180 and 1350135 transactions. The value of 1800180 transactions is in a no delay scenario and the value of 1350135 accounts for an assumed penalty if 25%. The raw data comes from the verify section of our table 2 and is calculated using the equation $TimePeriod \div MeanComputationTime$.

C. BLOCKCHAIN THROUGHPUT

We have calculated the size of the data-update transaction from CPS device to be 187 bytes in size and have done some theoretical calculations that estimate the number of transactions a blockchain system can handle with differing block sizes; this data is shown in Fig. 9. We have data points that show what the current two biggest blockchain solutions would be able to handle if all of the space in each of the blocks are used for the data update transactions, and the proposed scheme was implemented on them. If we assume that the CPS devices update their process data onto the blockchain every hour then this value can be taken as the number of CPS devices that the system can handle, since the graph y-axis is a value per hour. Bitcoin, shown in red on the graph, can hold 32085 transactions an hour and Ethereum, shown in green, and can currently handle 38502 transactions; and thus, this also means that they can handle this many CPS devices updating data hourly. These values are based on Bitcoins current block-size of 1 mb [38] every 10 minutes leading to a block capacity of 6 mb/hr and Ethereum's current block-size of 0.03 mb [39] every 15 seconds leading to a block capacity of 7.2mb/hr. Ethereum has a block-size that varies in size depending on the network congestion, so, this number will grow over time to accommodate more transaction throughput. Bitcoin and Ethereum have set block-size, and thus can be plotted on the graph and analyzed based on block-size as limiting factors. However, our implementation is a permissioned blockchain solution using Hyperledger Sawtooth that does not have a set block-size [32]. Sawtooth, being permissioned, has a much lower number of nodes that need to verify transactions and blocks, increasing the efficiency of the network. The permissioned nature also means that it will only receive transactions relevant to the CPTS network. The limiting factor in our system (Sawtooth) is computational power and network speed. For testing, our solution used a direct connection between the CPS devices and the blockchain so we cannot test for a realistic network delay. A real-world implementation of our scheme would require multiple parties running nodes, but our demonstrator system is only a single node to facilitate smooth demonstration; this leaves us only the ability to assess the computation power and speed theoretically.

VII. APPLICATIONS

In the following we will describe two use cases in which CPTSs driven by blockchain can find applications. Both have been co-created with teams of collaborators as mentioned in the Introduction I-C.

A. TRUST IN COMPONENT USAGE

We already introduced our industrial partner Riversimple in the Introduction I-C. A blockchain based CPTS can be used to realize Riversimple's aim of changing the ownership of key components for its vehicle prototype, the Rasa, to turn the Rasa and its core parts into services along the supply chain. For example, the fuel cell will use a CPS devise to record key usage data, like number of starts, total number of kWh it produced, etc. We assume that this CPS devise is equipped with a private public key pair and is able to sign its usage data in addition to any incoming data. Using our Scheme 1, the CPS devise will periodically record its data to the blockchain without actor facilitation: Every 10 min, the CPS device will request a PRV from the blockchain via its network connection. It then creates a message containing the PRV, its message counter and the key usage data, signs the message and sends it to the blockchain. The customer as well as any component manufacturer can then look up the data on the blockchain and be assured of the usage used for billing.

B. TRUST IN CROP DISEASE DETECTION AND COMPENSATION

Proposed research aims at supporting smallholder farmers in Columbia to use technology for early detection of pathogens in their crop and linking that to government providing pesticides and insurance companies reliably paying compensation. This would address many problems such farmers are currently facing, like farmers misidentify or completely miss plant diseases, overuse of broad spectrum pesticides with linked problems for human health and environment, as well as a genuine mistrust between all parties involved.

FIGURE 10. An agricultural drone monitoring crop [40].

At its heart, the proposed solution employs CPTSs driven by blockchain in the following way: Many plant pathogens can be detected at an early stage from images of plant leaves. Recent agricultural drones can be used to monitor crops, as shown in Fig. 10, by taking regular high resolution images of the farm land. Such agricultural drones are enhanced with CPS devices that will tag images with appropriate extra information like time, geo-location etc. and provide appropriate network connection. We assume that such a CPS devise is equipped with a private public key pair, and can sign its data as well as any data it receives. Using our Scheme 2, the CPS devise will periodically record this data to the blockchain with actor facilitation: Assuming that the drone operates in remote rural areas, it will request a PRV from the blockchain prior to take off when connection to the blockchain can be established with actor facilitation. It will then reuse the PRV for the whole session. For each image taken by the drone's on board camera, the CPS device will hash the picture, and form a message consisting of the hash, the PRV, its internal message counter, and any other relevant information like time and geo-location. It will then sign the message, and store it in a buffer to be send to the blockchain upon return. In this way we obtain a CPTS which allows all untrusted users of this ecosystem to gain trust in those images and their data tags. Typical users in this scenario are the smallholder farmer, government agencies providing pesticides, insurance companies insuring the crop. The CPTS would secure the tagged data from the camera and record it on a trusted blockchain system for enabling a crop support ecosystem as described above.

VIII. CONCLUSION

This paper makes a contribution to identify genuine blockchain applications. Such applications should solve a business problem, that cannot be solved more efficiently with different technology, have an identifiable network of actors, assets and transactions, and have a need in trust of recorded transactions, which includes properties like immutability, finality, provenance and consensus. We introduce CPTSs driven by blockchain via two schemes, distinguished by the presence of actor facilitation. Both schemes were analyzed in depth, providing formal proofs using the Tamarin tool, along side informal arguments for protection against replay attacks, resistance to spoofing attacks, device authentication and integrity, device identification, and non-repudiation. We also analyzed the performance of security primitives of a prototype implementation of our schemes. We concluded by describing two applications of blockchain based CPTSs, providing two genuine blockchain use cases.

REFERENCES

[1] E. Lee, "The past, present and future of cyber-physical systems: A focus on models," *Sensors*, vol. 15, no. 3, pp. 4837–4869, 2015.
[2] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.
[3] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A roadmap toward the resilient Internet of Things for cyber-physical systems," *IEEE Access*, vol. 7, pp. 13260–13283, 2019.
[4] G. Na, H. Lee, and Y. Eun, "A multiplicative coordinated stealthy attack for nonlinear cyber-physical systems with homogeneous property," *Math. Problems Eng.*, vol. 2019, Aug. 2019, Art. no. 7280474.
[5] J. Ramírez Gómez, H. F. Vargas Montoya, and Á. León Henao, "Implementing a wormhole attack on wireless sensor networks with XBee S2C devices," *Revista Colombiana de Computación*, vol. 20, no. 1, pp. 41–58, 2019.
[6] M. Nuss, A. Puchta, and M. Kunz, "Towards blockchain-based identity and access management for Internet of Things in enterprises," in *Trust, Privacy and Security in Digital Business*, S. Furnell, H. Mouratidis, and G. Pernul, Eds. Cham, Switzerland: Springer, 2018, pp. 167–181.
[7] P. K. Gray. *Cyber-Physical Attacks are Finally for Real*. Accessed: Mar. 27, 2020. [Online]. Available: https://symantec-blogs.broadcom.com/blogs/feature-stories/cyber-physical-attacks-are-finally-real
[8] B. Genge, P. Haller, and A.-V. Duka, "Engineering security-aware control applications for data authentication in smart industrial cyber–physical systems," *Future Gener. Comput. Syst.*, vol. 91, pp. 206–222, Feb. 2019.
[9] K. M. Renuka, S. Kumari, D. Zhao, and L. Li, "Design of a secure password-based authentication scheme for M2M networks in IoT enabled cyber-physical systems," *IEEE Access*, vol. 7, pp. 51014–51027, 2019.
[10] J. Wang, F. Luo, Z. Zhou, X. Luo, and Z. Wang, "Optimistic fair exchange in cloud-assisted cyber-physical systems," *Secur. Commun. Netw.*, vol. 2019, pp. 1–14, Oct. 2019.
[11] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 410–426.
[12] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," 2017, *arXiv:1706.01730*. [Online]. Available: http://arxiv.org/abs/1706.01730
[13] K. Lewison and F. Corella, "Backing rich credentials with a blockchain PKI," Pomcor, Carmichael, CA, USA, Tech. Rep., 2016. [Online]. Available: https://pomcor.com/techreports/BlockchainPKI.pdf
[14] C. Lin, D. He, X. Huang, K.-K.-R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, Aug. 2018.
[15] J.-H. Lee, "BIDaaS: Blockchain based ID as a service," *IEEE Access*, vol. 6, pp. 2274–2278, 2018.
[16] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. García, "Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 501, 2020.
[17] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.
[18] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Proc. Int. Workshop Open Problems Netw. Secur.* Zürich, Switzerland: Springer, 2015, pp. 112–125.
[19] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Comput. Secur.*, vol. 78, pp. 126–142, Sep. 2018.

[20] B. Seok, J. Park, and J. H. Park, "A lightweight hash-based blockchain architecture for industrial IoT," *Appl. Sci.*, vol. 9, no. 18, p. 3740, 2019.
[21] A. Beckmann, A. Milne, J.-J. Razafindrakoto, P. Kumar, M. Breach, and N. Preining, "Blockchain-based cyber physical trust systems," in *IoT Security: Advances in Authentication*, M. Liyanage, A. Braeken, P. Kumar, and M. Yliantilla, Eds. Hoboken, NJ, USA: Wiley, 2019, ch. 14, pp. 265–277.
[22] Merriam-Webster. *Trust*. Accessed: Mar. 27, 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/trust
[23] Lexico. *Trust*. Accessed Mar. 27, 2020. [Online]. Available: https://www.lexico.com/definition/trust
[24] *Framework for Cyber-Physical Syst., Release 1.0*. National Institute of Standards and Technology, Gaithersburg, MD, USA, 2016.
[25] *Oyster Bay Systems*. Accessed Feb. 14, 2020. [Online]. Available: https://www.oysterbaysystems.com/
[26] *Riversimple Movement*. Accessed: Feb. 14, 2020. [Online]. Available: https://www.riversimple.com/
[27] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 653–659, 2017.
[28] L. Xu, L. Chen, Z. Gao, Y. Chang, E. Iakovou, and W. Shi, "Binding the physical and cyber worlds: A blockchain approach for cargo supply chain security enhancement," in *Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST)*, Oct. 2018, pp. 1–5.
[29] S. Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Mar. 27, 2020. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf
[30] V. Buterin. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Mar. 27, 2020. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper
[31] H. Fabric. (2019). *Fabric Specification*. Accessed: Feb. 19, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/
[32] H. Sawtooth. (2017). *Sawtooth Specification*. Accessed: Feb. 19, 2020. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/1.0.5/architecture.html
[33] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology*, C. Pomerance, Ed. Berlin, Germany: Springer, 1988, pp. 369–378.
[34] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 858–880, 1st Quart., 2019.
[35] K. Chatterjee, A. K. Goharshady, and A. Pourdamghani, "Probabilistic smart contracts: Secure randomness on the blockchain," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 403–412.
[36] T. T. Team. (2019). *Tamarin-Prover Manual*. Accessed: Feb. 14, 2020. [Online]. Available: https://tamarin-prover.github.io/manual/
[37] A. Beckmann and A. Milne. (2020). *Sources for Tamarin Proofs*. [Online]. Available: http://beckmann.pro/TamarinProofs.zip
[38] *Block Size Limit Controversy*. Accessed: Feb. 24, 2020. [Online]. Available: https://en.bitcoin.it/wiki/Block_size_limit_controversy
[39] *What's the Maximum Ethereum Block Size*. Accessed: Feb. 24, 2020. [Online]. Available: https://ethgasstation.info/blog/ethereum-block-size/
[40] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "UAV-based crop and weed classification for smart farming," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3024–3031.

ALEXANDER J. M. MILNE received the degree in computer science from Swansea University, Wales, U.K., in 2018. He is currently pursuing the degree with Swansea University doing a KESS II funded masters by research on blockchain, where he has built a generic framework for tracking assets and storing their data securely on the blockchain. He worked at Swansea University on a CHERISH-DE funded project in collaboration with Oyster Bay Systems. This project was to implement a prototype blockchain system for tracking a specific simple asset. His research interests include Blockchain technologies, security and protocols, and logic.

ARNOLD BECKMANN received the Ph.D. (Habilitation) degree in mathematics from the University of Münster, Germany. He gained Postdoctoral experiences at the University of Oxford, the University of California at San Diego, and the Vienna University of Technology. He is currently a Professor of computer science with Swansea University. He conducts research in foundations of computer science, based on mathematical logic and theoretical computer science. He has developed a profile for transferring his expertise to applications. He is also a Founding Member of the Swansea Blockchain Laboratory, and involved in several projects that explore the application of blockchain technology to real world problems.

PARDEEP KUMAR (Member, IEEE) received the B.E. degree in computer science from Maharishi Dayanand University, Haryana, India, in 2002, the M.Tech. degree in computer science from Chaudhary Devilal University, Haryana, in 2006, and the Ph.D. degree in ubiquitous computing from Dongseo University, Busan, South Korea, in 2012. He is currently a Lecturer/Assistant Professor with the Department of Computer Science, Swansea University, Swansea, U.K. He has published more than 45 research articles and granted three patents. His research interests include security in sensor networks, smart environments, cyber physical systems, body area networks, the Internet of Things, and 5G networks.