From Hardware-Software Contracts to Industrial IoT-Cloud Block-chains for Security

Dimitrios Bakoyiannis
Elec. & Comp. Engin. Dept.
Hellenic Mediterranean Univ.
Iraklio, Crete, Greece
d.bakoyiannis@gmail.com

Othon Tomoutzoglou
Elec. & Comp. Engin. Dept.
Hellenic Mediterranean Univ.
Iraklio, Crete, Greece
otto@hmu.gr

George Kornaros
Elec. & Comp. Engin. Dept.
Hellenic Mediterranean Univ.
Iraklio, Crete, Greece
ORCID: 0000-0002-2371-0633

Marcello Coppola
STMicroelectronics
Grenoble, France
marcello.coppola@stm.com

2021 Smart Systems Integration (SSI) | 978-1-6654-4092-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/SSI52265.2021.9467030

***Abstract*—In the era of smart factories, to embrace IoT devices attached to physical assets, we need to guarantee control and complete confidence in how the data they share are used. This work introduces hardware mechanisms to ensure security in terms of secure key and signature storage through RFID/NFC secure modules and an IoT infrastructure communicating over LoRaWAN in conjunction with Hyperledger Fabric for traceability and immutability. A practical implementation is presented and evaluated with an average throughput of more than 70 transactions/sec for 16 peers.**

Keywords—LoRaWAN, IIoT blockchain, Hyperledger Fabric, RFID/NFC secure elements

I. Introduction

The proliferation of smart and connected networks of manufacturing things (e.g., materials, sensors, equipment, people, products, and supply chain) has raised the need for advanced methods to ensure trusted data integration, sharing and communication in Industrial IoT (IIoT) environments [1][2][9]. Despite profound advantages of cloud technology for IIoT ecosystems, the centralized nature of cloud services lacks transparency and trust, and smart factories embracing IoT devices and communications do not have control and complete confidence in how the data they share will be used. With IIoT driving unprecedented disruption in manufacturing, security becomes a first-order constraint in designing IoT infrastructures and devices. Today, IoT resource constrained devices may be equipped with hardware security elements to provide hardware support for cryptographic operations and tamper-proof memory for the secure storage of cryptographically sensitive data and code (e.g., authentication IDs and cryptographic keys). Integrated IoT devices with cyber-physical systems (CPS) in industrial equipment need to have a unique identity and guarantee untampered data over secure communications for immutable and auditable services.

To enhance tamper-proof data exchange among IIoT devices and cloud services, in this work we present a Distributed Ledger Technology (DLT)-based framework, which integrates HyperLedger Fabric with IIoT devices, connected over a LoRaWAN-based secure infrastructure. Considering the vastly varying devices involved in IIoT era, to achieve absolute decentralization using blockchain can be cumbersome. IIoT devices have resource constraints in power and computation, and can hardly accommodate for a DLT instance or engage in validating new blocks to reach consensus for the blockchain[3][10]. To address such challenges, we propose the application of smart contracts to leverage the immutability of the blockchain for generation of real-time access control lists that regulate and describe access policies to device resources. The key advantage of employing Hyperledger Fabric is that it offloads most communication and processing cost from the lightweight IoT to Hyperledger peers.

The structure of this paper is organized as follows. Section II discusses background concepts and related work. Section III introduces authentication methods for IIoT. Section IV presents the LoRaWAN IoT infrastructure combined with Hyperledger Fabric and section V delivers measurement results. Finally, section VI concludes this work and suggests future research directions.

II. Background and Related Work

Blockchain technology has introduced a new paradigm to facilitate message exchange in a decentralized way while promising to increase the efficiency of existing infrastructures [4][5] and scalable IoT management[12]. Essentially, it is a distributed ledger maintained by several network nodes which are mutual distrust while they can reach an agreement based on a consensus protocol, e.g., proof-of-work and proof-of-stake. The advantages of blockchain mainly involve *traceability* and *correctness*. Blockchain is a transparent data architecture so that each node can trace and verify the correctness of the data. In addition, the stored data are hard to tamper since they are organized as the special structures (e.g., hash chain), which ensures immutability and irreversibility. A participating IoT device with a unique digital identity performs a globally resolved transaction with each new verified transaction block linked with the previous recorded in the DLT. Maintaining the keys on the device can present an unacceptable security risk of key leakage unless the device utilizes a secure element, e.g., a trusted platform module (TPM), or embrace a proxy solution to act as a guardian for the keys.

In an IoT environment to avoid spoofing attacks and adversaries that masquerade as a sensor node, several authentication protocols have been proposed mostly based on lightweight mutual authentication methods [8][14], or ensuring that critical device firmware runs inside a Trusted Execution Environment (TEE)[15], which is a virtually secure area inside a main processor TEEs. In this scope, the ARM TrustZone provides architectural support to isolate security-critical services by protecting data and code inside trusted enclaves. Despite such hardware and protocol mechanisms, IoT networks security issues keep increasing, comprised by, end device attacks, network protocol, communication channel, denial-of-service and software attacks. Henceforth, combining IoT technology with blockchain offers immense benefits for establishing a trustworthy information sharing service that ensures data is immutable and tractable, thus providing answers to issues such as IoT data authenticity, reliability, scalability, and privacy[11].

Recently, in the context of IoT, practical implementations have been reported, which combine Hyperledger Fabric and ARM Trustzone in order to ensure secure execution of smart contracts[6]. Contrary to this direction towards running the Fabric private chaincode (FPC) in lightweight devices, we settle on the security model endorsed by LoRaWAN, which is further extended to securely store device and application IDs inside RFID/NFC. The RFID/NFC is capable to interact with a mobile phone app to configure the logistics codes. Key management mechanism of LoRaWAN environments with the support of blockchain technology has recently been

proposed[13]. However, the DLT is employed only for LoRaWAN keys, while we further propose and evaluate two use-cases which exploit actual sensor data.

Additionally, blockchain and smart contracts have also been proposed for a firmware update scheme for autonomous vehicles to ensure the authenticity and integrity of software updates[7], though the end-nodes utilize IEEE 802.11 wireless communications. Further, our proposed blockchain solution endorses event-based notification to enable OEM firmware updates to automatically advertise to connected vehicles.

III. Multi-Trust Authentication

To guarantee a trustworthy infrastructure, an IoT device needs to have a unique digital identity authenticating itself to the service it is part to. This IoT device may transmit a multitude of different sensor data (e.g. humidity, temperature, vibration, sound, image), which may provide different types of services. Additionally, different access levels commonly may be required for the software running on a IoT device, for instance, to communicate the generated sensor data streams, or to enable re-programming of sensor parameters. Further, as the developers update the firmware frequently to fix bugs, to update protocols or application features, new releases of full firmware or partial segments need to be securely installed during the lifetime of a device.

To protect against internal threats (e.g., software executing on the microcontroller itself) and external attacks (e.g., attacks triggered by external tools such as debuggers or probes, trying to access the device), modern IoT devices integrate hardware protection methods such as firewalls and isolated memory compartments for keys, device identifiers and certificates. Most microcontrollers embed unique IDs programmed by their silicon vendors for binding to a specific device, which can be used in conjunction with cryptographic protocols. For example, the NXP i.MX RT1064 includes such a 64-bit ID and STM32L5 series use a 96-bit ID. However, in software-based security mechanisms the keys are stored in the non-volatile memory (NVM) of the devices, which can be prone to attacks. A possible hardware countermeasure solution for identification and authentication in IoT, is hardware security module (HSM), or secure element [1] that is suggested to enhance security through secure certificate storage and key management services.

To achieve mutual security handshake and trust between two devices via a secure on-demand wireless connection, RFID/NFC technology is traditionally used. Near-field communication (NFC) tags such as the STM25 provide extra advantage such as tamper detection as well as strong cloning-prevention, data-protection, and user-privacy features[16]. By using a unique electronic ID-tagging and anti-tampering mechanism, NFC tags can be used in IIoT applications that require authenticity and traceability of products/data while enabling secure device configuration, firmware update, cryptographic keys setup and access. By using a UID of the RFID/NFC tag with a digital signature, we ensure information traceability by writing all stages of circulation process to blockchain. This two-way authentication method gives to end user a convenient way to access and view IIoT related data, so counterfeiting can be detected immediately. Blockchain technology can guarantee data tampering protection, but it cannot guarantee *authenticity* and *reliability* of data source. Thus, as shown in the process in Fig.1, the combination of RFID/NFC with Digital Signature capabilities and the blockchain technology enable uploading of true data to the ledger, guarantees for data tamper resistance, along with data authenticity and reliability of the data source. Essentially, methods and use-cases we employed to authenticate industrial IoT devices (and subsequently get sensor data or transmit data e.g., for reaction or FOTA updating) include (i) Over-the-air (OTA) firmware authentication via RFID/NFC, (ii) remote (through cloud-based dashboard) firmware verification and authentication.

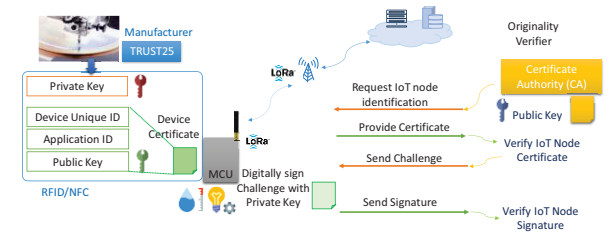

Fig. 1. Cloud-based authority verification of the IoT device identities and cryptographic credentials that are stored in the internal RFID/NFC ST25 module with TRUST25 and TPM of the IoT device.

IV. LoRaWAN and Hyperledger Fabric for IIOT

IoT devices can boost the productivity of an industrial process by providing feedback through sensor monitoring (e.g. predictive maintenance) or by adopting actuators for control and automation improvements. In this context, we implement an industrial LoRaWAN monitoring infrastructure which makes it feasible for monitoring services and applications to interact with industrial IoT End Node devices. In this implementation, we integrate the Hyperledger Fabric blockchain technology; by exploiting its capabilities for immutable records and decentralization we ensure data integrity and avoid single point of failure when accessing those records in a distributed cloud environment. Additionally, Hyperledger Fabric is a private network which is a desired feature for industries where data must be available only for predefined trusted entities.

A. IIoT End Nodes

Data are acquired from connected sensors and transmitted to STM32 MCU through the wired interface, then processed and organized into standard LORA packets. A data stamp is extracted through hash calculation and signed using ST25 & STM32. Then, the STM32 uploads the signed data stamp (and a data time-stamp if needed) to the blockchain through using the LoRaWAN infrastructure and a bridging-proxy service, called *Isthmos* hereafter. At the same time Isthmos uploads the assembled original data to the centralized server. Thus, when used in tracking applications, these IIoT nodes enable the correct processing of product constraints such as temperature, humidity etc., during the shipment process. Last but not least, it also prevents theft and falsification.

B. The Networks of the Equation

The LoRaWAN network is one part of the equation where IoT devices connect to a gateway in order to forward (uplink) data to a LoRaWAN server or receive (downlink) data from the latter. The other part of the equation is the Hyperledger Fabric network that is used for recording data from the IoT devices, which are consumed (read) by services e.g., for monitoring purposes. The LoRaWAN server is the point that allows interaction with services and applications that are not part of the LoRaWAN network. This server does not integrate any built-in solution to allow direct interaction with the

[1] STSAFE-A110 secure element is a tamper-resistant secure element (Hardware Common Criteria EAL5+ certified) used to host X509 certificates and keys and perform verifications that are used for firmware image authentication during Secure Boot and Secure Firmware Update procedures.

Hyperledger Fabric network, thus, we introduce *Isthmos* (as shown in Fig. 2), a service that acts as a bridge to make communication between the two networks achievable.

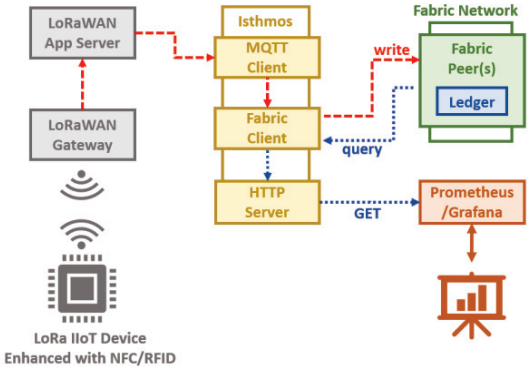

Fig. 2. A simplified architecture of the network infrastructure.

C. Isthmos

Isthmos is a multi-functional application developed in NodeJS for bridging the LoRaWAN and Hyperledger Fabric networks as well as allowing other services to extract records of the blockchain ledger for monitoring/control purposes. A main component of Isthmos is the integration of a Fabric client which uses the Fabric SDK to access the entities of the Fabric network. The Fabric SDK allows the following operations: (i) register/enroll with a Fabric built-in certificate authority (CA) for acquiring cryptographic material to secure further communications; (ii) invoke a peer's chaincode (smart contract) in order to either write a new record to the ledger or query one; (iii) connect with a peer to register for events of committed transactions (records) or chaincode events. Another aspect of Isthmos is the integration of an MQTT client which subscribes to an MQTT broker of the LoRaWAN server for receiving published uplink messages derived from IoT devices. These uplink messages contain data that must be processed by Isthmos which in certain cases calls the Fabric client to write those data to the ledger.

Finally, Isthmos is equipped with a native HTTP server which listens to requests from monitoring services. Processing such requests is a matter of calling the Fabric client to acquire data from the ledger that must be provided as a response to an HTTP request. We have chosen to utilize the Prometheus monitoring system which is configured to request (HTTP) data from Isthmos at fixed time intervals. Further, we utilize the Grafana tool which, among others, has native compatibility (plugin) for collecting and visualizing data from Prometheus services.

D. Hyperledger Fabric

The Hyperledger Fabric allows the creation of highly flexible and modular blockchain networks regarding the members that are part of the network, the policies that dictate access rights to specific members, the level of security and authentication, the database to use as well as the number of channels that isolate information in different ledgers. The implemented Fabric network is structured as a consortium where different organizations take part in it (see Fig. 3). Separating the network in organizations simulates real cases where each organization has to satisfy its own interests regarding the way to interact with the ledger. Additionally, administrators of the blockchain network may apply different access policies per organization. In the context of flexibility, we have chosen to create uniform organizations, that is, each organization includes one orderer node, one CA and an equal number of peers per organization. The orderers of the network use the Raft consensus method to collaborate and agree for the transaction ordering.

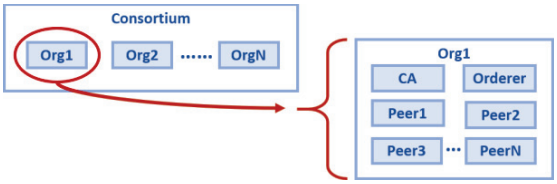

Fig.3 Entities of an organization in a Hyperledger Fabric consortium.

E. Use cases

As described above, a combination of the LoRaWAN and the Fabric network is used to access and manage IoT devices of an industrial environment. Another, challenging aspect to consider when setting up a network of IoT devices is the required maintenance that must take place, that is, provide the means to update a device's firmware to essentially make a device immune to security threats and less prone to errors and malfunctions that may arise. As a result, we consider two use cases to study: (i) store IoT device data (e.g. sensor values) at the ledger to create a history of data records that are used for monitoring, and (ii) record update information metadata at the ledger that can be used to manage the firmware update procedure of IoT devices of the LoRaWAN network.

In the first use case, a number of IoT devices forward sensor data or device information data, namely the current configuration and status of the device. Those data are packaged in an uplink message, transmitted to a LoRaWAN gateway and then forwarded to a LoRaWAN server for processing. Since the IoT device data need to be consumed for monitoring, they are published by the LoRaWAN server and received by Isthmos, which in turn calls its Fabric client to create a transaction for immutably storing those data to the ledger of the peers. At the same time, an external Prometheus service access the native HTTP service of Isthmos to request the last record of data or a batch of records of data of a single or multiple IoT devices. For every request that Isthmos receives from Prometheus, it makes a call to its Fabric client to query the ledger for acquiring the demanded records. Once those data are handed to Prometheus, they are also made available to Grafana for visualization.

In the second use case, the combination of the Fabric network and Isthmos is used to manage an update procedure and deliver a new firmware to a group/family of IoT devices. Managing a firmware update is a matter of combining two different types of information. One is from the perspective of the OEM that must provide the available version of firmware for a family of IoT devices and the other is from the perspective of the IoT device that must signify its currently installed firmware version. We use metadata files preferably in JSON format to constrain all the information required for a firmware update. An OEM may include in its metadata the family of devices that the firmware refers to, the version of the firmware, hashes for integrity checks and size information. The metadata of an IoT device may include its ID, the installed version of firmware, hashes and size information.

The update procedure starts on Isthmos which registers for transaction events from the Fabric network. At a second step, an OEM writes a record of firmware metadata at the ledger to inform about a new available firmware image. Isthmos receives an event and queries the ledger to acquire and examine the OEM firmware metadata. Since it acts on behalf of a hardware restricted IoT device it fetches a firmware image from a repository server and makes integrity and version checks based on the OEM firmware metadata. It then uses its MQTT client to publish the firmware image in chunks at the LoRaWAN server which in turn stacks the chunks of the

image in a queue so that they can be transmitted as downlinks to the IoT device. Once the latter receives and installs the new firmware, it is responsible to send an uplink message to signify the successful installation. On being notified, Isthmos makes an IoT device metadata record at the ledger, which is necessary for keeping track of the installed firmware version, as well as providing an anti-rollback protection mechanism.

F. Channels

In Hyperledger Fabric a channel is a private "subnet" of communication between two or more specific network members, for the purpose of conducting private and confidential transactions. Administrative entities of the Fabric network may create multiple channels and restrict access for specific members in each of them. Another advantage of adopting multiple channels is that each channel has its own ledger which is isolated from other channels. This is practical for separating different types of records to different ledgers which introduces better management and may allow access to only an interested group of members (e.g. department of an industry). Additionally, transactions inside a channel are validated sequentially which may introduce bottleneck conditions when processing multiple transactions, while transactions of one channel are validated in parallel with transactions of other channels. Thus, by fragmenting the transactions taking place in multiple channels, increases the overall transaction throughput.

V. Evaluation Results

In order to evaluate the transaction rate, we created and tested different Fabric network configurations based on the number of organizations, peers and channels. The tested combinations were for networks of 1, 2, 4 and 8 organizations where in every case each organization is equipped with two peers. These combinations were repeated for 1, 2, 3 and 4 channels, which leads to a total of 16 different tested network configurations. To test each network, we introduced a benchmark client application, specifically developed for our needs, which connects to a peer and starts invoking 2000 transactions sequentially for each available channel. In every test there was one benchmark client for each peer. All tests were performed in a single host machine equipped with an Intel Xeon E3-1246 3.50GHz processor and 16 GB of 1600 MHz DDR3 memory.

Fig. 4. shows the transaction rate achieved when scaling the number of peers in a network and when scaling the number of channels. Transactions in a channel, are processed sequentially, thus, when the number of transactions outreach a certain threshold the validation process becomes a performance bottleneck even with low CPU usage. This behavior can be seen for all networks of a single channel, where the network introduces a transaction rate degradation beyond eight peers. When introducing multiple channels, the validation process for each channel is executed in parallel with other channels, and thus, fragmenting the transactions in multiple channels increases the transaction rate. This is clearly observed in networks of two peers where the transaction rate for 1 channel is 33 transactions per second (TPS) and gradually reaches a maximum value of 68 TPS for 4 channels. Increasing the number of channels though, causes a bottleneck after a certain threshold. In the cases of 4, 8 and 16 peers it is obvious that there is no gain by increasing the number of channels to more than 3. The bottleneck in this case is mainly due to the CPU utilization since the Fabric network occupies all the available processors for the parallel execution.

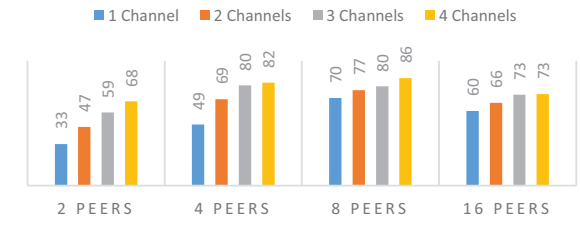

Fig. 4 Average TPS values for different Fabric network configurations

VI. Conclusions

In this work, we presented a cooperative scheme of enhancing IIoT device authentication via hardware-based secure modules, together with a decentralized authentication and integrity assurance framework for IIoT devices using a private Hyperledger Fabric. A practical framework of integrating IIoT sensors over LoRaWAN with blockchain has been developed. We showed how increasing the number of peers, while fragmenting the network in multiple channels, can affect the transaction rate. In the future, we intend to extend the Hyperledger Fabric implementation to smart contracts in the validation of multi-protocol sensor data and of more dynamic approach of policy control.

Acknowledgment

The research leading to these results received funding from the European Union (EU) project AVANGARD (No 869986).

References

[1] Fernández-Caramés, T. M. and Fraga-Lamas, P., "A Review on the Application of Blockchain for the Next Generation of Cybersecure Industry 4.0 Smart Factories," https://arxiv.org/pdf/1902.09604.pdf

[2] O. Vermesan et al., "New waves of IoT technologies research – transcending intelligence and senses at the edge to create multi experience environments," in Internet of Things - The Call of the Edge – Everything Intelligent Everywhere, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. DK: River Publishers, 2020, ch. 3, pp. 17–184.

[3] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight IoT clients," IEEE Internet of Things J., vol. 6, no. 2, pp. 2354–2365, 2019.

[4] Q. Zhu et al., "Applications of Distributed Ledger Technologies to the Internet of Things: A Survey", ACM Comp.. Surv. 52, 6, 2019.

[5] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," IEEE Commun. Mag., vol. 55, no. 12, pp. 119–125, Dec. 2017.

[6] C. Muller, et al., "TZ4Fabric: Executing Smart Contracts with ARM TrustZone : (Practical Experience Report)," in Proc. Int'l Symposium on Reliable Distributed Systems (SRDS), 2020, pp. 31-40.

[7] M. Baza et al., "Blockchain-based firmware update scheme tailored for autonomous vehicles," in Proc. of IEEE Wirel. Comm. Netw. Conf., 2019, pp. 1–7.

[8] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," IEEE Trans. Wireless Commun., vol. 15, no. 1, pp. 357–366, 2016.

[9] X. Tong, Q. Liu, S. Pi, and Y. Xiao, "Real-time machining data application and service based on imt digital twin," Journal of Intelligent Manufacturing, vol. 31, pp. 1113–1132, 2020.

[10] C. Yu, X. Jiang, S. Yu, and C. Yang, "Blockchain-based shared manufacturing in support of cyber physical systems: concept, framework, and operation," Robotics and Computer-Integrated Manufacturing, vol. 64, p. 101931, 2020.

[11] B. Farahani, F. Firouzi, M. Luecking, "The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions", Journal of Net. and Comp. App., vol. 177, 2021, 102936.

[12] L. Tseng, et al., "Blockchain for Managing Heterogeneous Internet of Things: A Perspective Architecture," in IEEE Network, vol. 34, no. 1, pp. 16-23, January/February 2020.

[13] V. Ribeiro, et al., "Enhancing Key Management in LoRaWAN with Permissioned Blockchain", vol. 20, no. 11, p. 3068, May 2020.

[14] D. Mbakoyiannis, O. Tomoutzoglou, and G. Kornaros, "Secure Over-the-air Firmware Updating for Automotive Electronic Control Units", Proc. of 34th ACM/SIGAPP Symp. on App. Comp., pp. 174-181, 2019.

[15] G. Kornaros, et al., "Towards Holistic Secure Networking in Connected Vehicles through Securing CAN-bus Communication and Firmware-over-the-Air Updating", Journal of Systems Architecture, vol. 109, pp. 101761, 2020.

[16] STM, "Developing supply chain confidence with TruST25™ digital signature in RFID/NFC tags", Technical article, TA0358. [Online]. Available: www.st.com/content/st_com/en/landing-page/trust25-digital-signature-from-factory-to-consumer.htm