Fog Computing and Blockchain based Security Service Architecture for 5G Industrial IoT enabled Cloud Manufacturing

Tharaka Hewa, *Student Member, IEEE,* An Braeken,
Madhusanka Liyanage, *Senior Member, IEEE*, Mika Ylianttila, *Senior Member, IEEE*

***Abstract*—Recent evolution of the Industrial Internet of Things (IIoT) empowers the classical manufacturing model with cloud computing integration for Industry 4.0. Cloud integration advances the capabilities of manufacturing systems with cloud-based controlling and real-time process monitoring, which is renowned as Cloud Manufacturing (CM). However, cloud integration exposes the entire manufacturing ecosystem to a new set of security risks and increments in end-to-end latency. Moving security services towards the edge eradicates message routing latency towards the cloud and eliminates the central point of failure while leveraging the entire system's performance. We propose a blockchain and fog computing enabled security service architecture that operates on fog nodes at the edge of manufacturing equipment clusters. The proposed service facilitates CM equipment authentication and Equipment-Cloud channel privacy protection while preserving anonymity and unlinkability over the blockchain. We implemented the proposed architecture with Hyperledger Fabric and compared the performance advantage over the state of the art solutions.**

***Index Terms*—Cloud Manufacturing, Fog Computing, Blockchain, Smart Contracts, 5G, IoT, Security, Hyperledger**

I. Introduction

Cloud Manufacturing (CM) is a vibrant and novel manufacturing paradigm, which transforms the classical manufacturing model into a customer-centric and service-oriented business architecture [1]. It allows product developers to control the remote manufacturing plants over the Internet. CM offers several benefits such as the possibility to build cost-effective manufacturing plants, enable access to less-expensive manpower and manufacturing materials and the possibility to manufacture the products closer to the consumer market.

With the evolution of IoT, the components of the production line, such as actuators, and sensors transformed into cyber-physical manufacturing systems [2], [3] connected to the cloud over the internet. The cyber-physical manufacturing systems have smart capabilities as well as inherit the security threats. With the evolution of IoT, the components of the production line, such as actuators, and sensors transformed into cyber-physical manufacturing systems [2], [3] connected to the cloud over the internet. The cyber-physical manufacturing systems have smart capabilities as well as inherit the security threats [4] prone to the IoT.

In [5] and [6], authors highlight the key information security challenges in the CM context and elaborate on the requirement of trust establishment. In addition, the CM system is suffering from high latency due to cloud integration [7]. In [8], a fog-based architecture has been proposed for a smart manufacturing environment addressing the latency issues of cloud-based architectures. In [9] another fog based authentication and key agreement protocol has been proposed, which was limited to elliptic curve operations and does not require user interaction on the IoT devices. Sciancalepore et al. [10] proposed ECQV implicit certificate based key management protocol for mobile and IoT. Shen et al. [11] proposed a blockchain-assisted IoT device authentication scheme based on identity-based signature schemes. In [12], a symmetric key-based scheme for the fog architecture has been proposed, satisfying also besides mutual authentication, anonymity and unlinkability.

In addition, blockchain has an immense potential to leverage CM security. With the distributed smart contracts, CM IoT nodes reach security services faster than the services hosted in the cloud. The immutable ledger ensures accountability and non-repudiation of the transactions committed. Bouachir et al. [13] highlight the applicability of blockchain and fog computing for the enhancement of security and service values of IIoT applications. Gadekallu et al. [14] present a review on the different applications of blockchain-enabled Edge of Things (EoT). The significance of blockchain for the smart manufacturing and Industy 4.0 is presented in [15]–[17] We also distinguish the scheme of [18] in which the blockchain is used to create a fully distributed access control system for IoT. The applications of blockchain for smart manufacturing oriented IoT security is presented in [19], [20]. Dorri et al. [21] proposed a scalable and blockchain-based security service for resource restricted IoT networks. Kumar et al. [22] proposed a blockchain based framework which delivers a set of value additions to the IIoT networks, including IIoT security. Wang et al. [23] proposed a lightweight certificateless authentication scheme for IIoT.

From the literature, we identified three key security challenges of CM, i.e., 1) Establish trust in manufacturing/monitoring equipment to ensure the product authenticity

Tharaka Mawanane Hewa and Mika Ylianttila are with the Center for Wireless Communications, University of Oulu, Finland. e-mail:{firstname.lastname}@oulu.fi

An Braeken is with the Vrije Universiteit Brussel, Brussels, Belgium. e-mail: an.braeken@vub.be

Madhusanka Liyanage is with the School of Computer Science, University College Dublin, Ireland and the Center for Wireless Communications, University of Oulu, Finland. e-mail: madhusanka@ucd.ie, madhusanka.liyanage@oulu.fi

This work is supported by Academy of Finland in 6Genesis Flagship (grant no. 318927) project.

TABLE I: Important notations and acronyms

Acronym	Definition
$5G$	5th Generation
$5GTN$	5th Generation Test Network
CA	Certificate Authority
CM	Cloud Manufacturing
CSP	Cloud Service Provider
DH	Diffie-Hellman
ECC	Elliptic Curve Cryptography
$ECDHP$	Elliptic Curve Diffie Hellman Problem
$ECDLP$	Elliptic Curve Discrete Logarithm
$ECQV$	Elliptic Curve Qu Vanstone
$ECIES$	Elliptic Curve Integrated Encryption Scheme
EoT	Edge of Things
$Gbps$	Giga bit per second
$IIoT$	Industrial Internet of Things
$IPFS$	Inter Planetary File System
IoT	Internet of Things
PKI	Public Key Infrastructure
ZKP	Zero Knowledge Proof

and manufacturing lifecycle consistency, 2) Ensure privacy over the internet to secure sensitive manufacturing instructions/monitoring data compromise, 3) Scalability requirements of the security services to cope with future expansions. However, none of the existing CM solutions are able to address all of the above key security challenges.

A. Motivation and contributions

To resolve the above-identified issues, we propose a novel fog computing and blockchain-based security services architecture with the following features.

1) Dynamic Elliptic Curve Qu Vanstone (ECQV) certificates and Elliptic Curve Integrated Encryption Scheme (ECIES) enabled blockchain-based security service for trust establishment.
2) Diffie-Hellman (DH) key exchange for the establishment of a symmetric key between IoT-Fog-Cloud channel to encrypt the manufacturing related message traffic.
3) Non-interactive Zero Knowledge Proof based verification in the security service to ensure anonymity and unlinkability on dynamic identities stored in the ledger.
4) Extended storage integration to the distributed ledger to improve the storage scalability for the facilitation of massive IoT quantity to use the proposed security service.

To the best of the authors' knowledge, no other blockchain-based security schemes have been proposed for fog based cloud manufacturing process in which the remote IoT device wants to anonymously utilize the resources of the fog to establish a secure connection with the cloud. We evaluate the proposed scheme in terms of storage overhead, search latency, and end-to-end latency with a comparison of the state-of-the-art works. Table I includes the important acronyms and notations.

II. Preliminaries

A. Elliptic Curve Cryptography (ECC) and Elliptic Curve Integrated Encryption Scheme (ECIES)

The most lightweight public key-based operations are realized with Elliptic Curve Cryptography (ECC), which is defined on the algebraic structure of elliptic curves (EC) over finite fields F_p generated by a generator point G. The two major operations in ECC are EC point addition $P_1 + P_2$ and EC point multiplication rP with a scalar r. The security of ECC relies on two computational hard problems, the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the Elliptic Curve Diffie Hellman Problem (ECDHP).

B. Elliptic Curve Integrated Encryption Scheme (ECIES)

ECIES is a very efficient encryption algorithm. For encrypting the message M to the receiver with public key Q_R, the sender generates a random value r and computes $A = rG$. The symmetric session key K is now defined by $K = rQ_R$, which can also be derived by the receiver $K = d_R A = rQ_R$, who knows the private key d_R for which $Q_R = d_R G$.

C. Schnorr signature scheme

In order to sign a message M, the user with key pair $(d_n, Q_n = d_n G)$ chooses a random value r and computes $R = rG$. Next it derives $h = H(M, R)$, where $H(.)$ is a one-way collision-resistant hash function. The signature is then defined by $s = r - hd_n$. The user makes the message M, together with R, s public. Any other party can now verify that the message M is signed by the user with public key Q_n, by checking the equality $sG = R - H(M, R)Q_n$. We also shortly denote this process by

$$M_{s_{d_n}} = \{M, R, s\} \tag{1}$$

for the signature generation and $M_{s_{Q_n}} = \{Y, N\}$ for the signature verification with P or N/P as outputs.

D. EC non interactive ZK proof based on Schnorr signatures

In this proof, we need to assume that prover and verifier agree on the EC, the generator G and one additional EC point P. The goal of the proof is to convince the verifier that the prover possesses b, given $B = bG$ and without sharing additional information on b.

To this end, the prover generates a random value r and computes $A = rG$. Next it defines $c = H(xP, rP, A)$ and $s = r + cx$. The proof consists of the set of values $\{s, xP, rP, A\}$.

Upon arrival of the proof, the verifier first computes $c = H(xP, rP, A)$ and checks if both equalities $sG = A + cB$ and $sP = rP + cxP$ are valid. If so, the verifier is convinced.

III. Proposed architecture

Machine to cloud connectivity over the internet is a key feature of almost all cloud manufacturing ecosystems. The manufacturing equipment should be trusted within the ecosystem to ensure the manufacturing life cycle is consistent and intellectual properties are preserved.

In the proposed architecture, the manufacturing equipment utilizes dynamic ECQV certificates and is defined as a trusted IoT node once it has a valid ECQV digital certificate registered on the blockchain for each session. Furthermore, the proposed

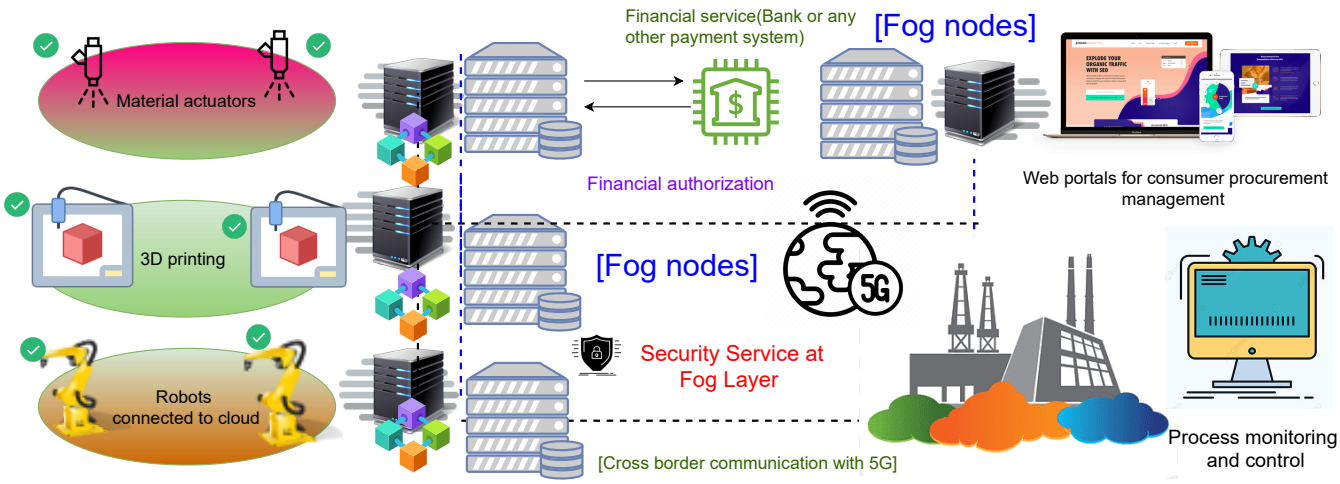


Fig. 1: Overview of the proposed architecture

architecture facilitates to establish a symmetric key between the manufacturing equipment and the cloud to encrypt the manufacturing operation related messages (eg. manufacturing instructions sent from cloud to the actuators, sensor data sent from equipment towards the cloud).

In the architecture, it is proposed to make the ECQV certificate generation and symmetric key establishment as a dynamic operation in order to minimize the security risk in encryption (symmetric) and authentication (private key of ECQV certificate) key compromise.

We propose to integrate the blockchain to leverage the ECQV certificate management operation and IoT-Cloud key establishment process with lower latency when compared with the cloud-based service architecture. Considering the key related works [24]–[27], we envisioned an architecture of a generic CM system with the components including,

Cloud Service Provider (CSP): which holds digital and intellectual product designs, manufacturing instructions, and performs process monitoring.

Manufacturing equipment: which are the actuators and sensors to perform the manufacturing process. These entities are assumed to be connected to the network (Wi-Fi or 5G) via integrated IoT nodes. The manufacturing equipment is referred to as IoT nodes in the proposed architecture.

Fog nodes: which connect the manufacturing equipment and underlying IoT nodes to the cloud. Furthermore, it is assumed that the fog nodes are operating as blockchain nodes to perform security related operations. It is possible to have multiple fog nodes with independent manufacturing equipment groups which are connected to the same CSP.

Consortium blockchain: which operates as a consortium consisting of a CSP and independent manufacturing equipment groups. It is assumed that each fog node and CSP operate as full blockchain node, which performs transactions committed to the ledger and contributes to the consensus process. Figure 1 illustrates a high-level overview of the proposed architecture.

Since a production line of CM contains numerous cyber-physical manufacturing equipment, the proposed architecture is designed to onboard a scaled-up quantity of equipment. Especially, the storage growth of the blockchain incurs overhead on the fog nodes. To overcome storage growth overhead on the blockchain, we propose to offload the dynamic ECQV certificate data towards a distributed storage system to increase the storage scalability of the proposed architecture. For example, distributed storage systems such as InterPlanetary File Systems [28] provide optimization in repeated queries and the integrity of the stored data is ensured through hashing.

In principle, cloud manufacturing has been identified as a pay-as-you-go business model [1]. We envisioned that the proposed solution facilitates payment from the consumers for the security service. It is assumed that the smart contract integration with the off-chain banking service facilitates financial transactions for the registration of IoT nodes. Even though payment handling is beyond the scope of CM security, we briefly explain the flow of registration for the consistency of the proposed work. However, we did not evaluate the registration workflow since it is not directly related to the security of the proposed system.

A. Cloud, Fog and IoT node registration on blockchain

The CSP-CA has the public key $Q_c = d_c G$ and the private key d_c stored privately. The fog nodes register the address corresponding with $H(ID_f, d_f Q_c)$ on the blockchain, where d_f is stored privately in the fog node. ID_f is the identifier of fog node.

This step is important to populate the ledger with dynamic and unlinkable ECQV digital certificates in order to verify at each session $i = 1$ to $i = n$ against a payment. For the manufacturing equipment α, the customer generates a private value w and $ID - Cert_i$ (as indicated in Equation 2) for $i = 1$ to $i = n$. The set of certificate identifier records I_α are indicated in Equation 3.

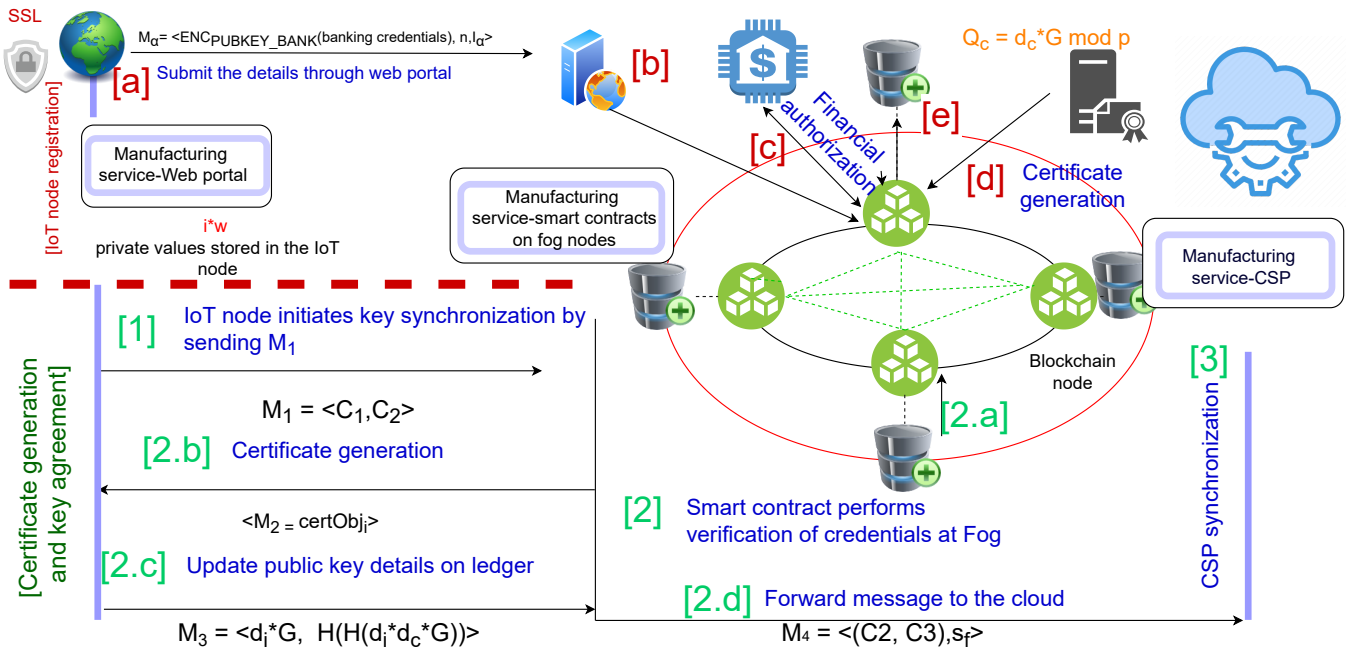
$$ID - Cert_i = H(ID_i, i * w)G \tag{2}$$

The set of certificate identifier records over the blockchain I_α are denoted as,

$$I_\alpha = \{ID - Cert_i \dots ID - Cert_n\} \tag{3}$$

Figure 2 illustrates the proposed architecture for registration and key exchange. At Step a, the customer submits the payment service details (eg. credit card no, Bitcoin or Ethereum

Fig. 2: Proposed architecture : Registration and key exchange



wallet information), I_α and expected number of dynamic IoT-CSP sessions (n) to the web portal. At Step b, the backend web server receives the request and at Step c, the request is forwarded to the blockchain. In the blockchain, Algorithm 1 is executed and it is assumed that the financial service request is initiated by the smart contract to pay the subscription fee.

Algorithm 1 Certificate generation smart contract in CSP-CA

Require: $M_\alpha = \langle A_\alpha, n, I_\alpha \rangle$
  $debitAmount = calculateAmount(n)$
  $response = verifyAccountFromPaymentService(A_\alpha, n)$
  if $response == success$ **then**
    while $i \leq n$ **do**
      $I_\alpha - i \leftarrow H(ID_i, i*w)G = ID - Cert_i$
      $r_c - i = generateRandomValue()$
      $Cert_i = H(ID_i, i*w)G + r_{c-i} * G$
      $r_{aux-i} = H(H(ID_i, i*w)G, Cert_i)r_{c-i} + d_c$
      $BC - Cert_i \leftarrow \langle ID - Cert_i, Cert_i, r_{aux-i} \rangle$
      $storageID \leftarrow addToStorage(BC - Cert_i)$
      $ledgerRecordID \leftarrow addToLedger(storageID)$
      $i \leftarrow i + 1$
      return $success$
    end while
  else
    return $failedPayment$
  end if

The Cloud Service Provider-Certification Authority (CSP-CA) has the public key $Q_c = d_c G$ and d_c stored privately. Upon successful response from the bank, the smart contract generates the certificate object and stores it in the extended storage with storage address pointer stored on the blockchain. In the certificate generation (Step d), the CSP-CA generates a random value r_{c-i} for each certificate generated from $i = 1$ to $i = n$. The CA generates the certificate $Cert_i$ defined as,

$$Cert_i = H(ID_i, i*w)G + r_{c-i} * G \quad (4)$$

The CSP-CA generates the individual auxiliary parameter r_{aux-i} for the generating certificates from $i = 1$ to $i = n$.

$$r_{aux-i} = H(H(ID_i, i*w)G, Cert_i)r_{c-i} + d_c \quad (5)$$

The record of the certificate i is indicated as $BC - Cert_i$,

$$BC - Cert_i = \langle ID - Cert_i, Cert_i, r_{aux-i}, Timestamp \rangle \quad (6)$$

The CSP-CA, which is a sub component of CSP is invoked by the local blockchain node in the CSP premises. However, the CSP-CA operational mode differs from centralized CAs since the CSP-CA does not handle the end to end certificate management operation.

Table II indicates the ledger record stored in the blockchain which is identified by $H(ID_i, i*w)G$. Table III indicates that the record is stored in the extended storage corresponding to the certificate i. The extended storage record links with the ledger record with $ExtendedStorageKey$. The $ExtendedStorageKey$ is the hash of the content of Table III. This setup offloads the storage overhead to the extended storage and increases the scalability of the ledger storage. Furthermore, the $ExtendedStorageKey$, which is ideally the hash of the record facilitates the blockchain to ensure the integrity of the off-chain stored record. Step e indicates the storage of the certificate object in the extended storage. At the end of certificate generation operation, two dynamic records are generated at each dynamic session i. Table II shows the record stored in the ledger and Table III shows the record stored in the extended storage for the dynamic session i.

The infrastructure for the off-chain storage can be a fog node or virtual machine which can be accessed by the smart contract with lower latency. However, in a real deployment, a virtual machine in the same wired/wireless network or local 5G operator domain will yield a lower latency in the off-chain storage access from the smart contract.

TABLE II: Certificate record stored in the ledger

Key of ledger record	Key of storage
$H(ID_i, i*w)G$	$\langle ExtendedStorageKey \rangle$

TABLE III: Certificate object stored in the extended storage

Element	Description
Key of storage	$\langle ExtendedStorageKey \rangle$
Certificate	$H(ID_i, i*w)G + r_{c-i} * G$
Counter data	$H(ID_i, i)$
Auxiliary data for private key generation	$H(H(ID_i, i*w)G, Cert_i)r_{c-i} + d_c$
Curve point for ZKP	P_i
ZKP data set	$\{s_i, x_iP_i, r_iP_i, A_i\}$

B. Session-based certificate activation and IoT-CSP channel key establishment

1) Manufacturing equipment initializes key establishment

The primary requirements of the proposed ecosystem include the certificate establishment and the IoT-CSP symmetric key establishment. Within each key exchange $i = 1$ to $i = n$, the IoT node generates the private key at each key exchange based on the auxiliary data stored in the ledger for key exchange i. Furthermore, the cloud node requires to be synchronized with the counter i and the dynamic identity ID_i without revealing the knowledge to the blockchain service and fog service layer. The IoT node sends M_1 to the fog node for the initiation of the key exchange.

The message M_1 consists of ciphertexts C_1 and C_2. C_1 includes the certificate identifier in the encrypted form and C_2 includes the tuple composed with the dynamic identity and counter in encrypted form.

The fog nodes registered in the ledger with public key $Q_f = d_fG$ and private key d_f is stored as private. C_1 requires to be decrypted at the fog node in order to query the certificate object stored in the ledger. Aligning to the ECIES, the random value r_f will be generated in the IoT node. The value $A_f = r_fG$ and the symmetric key K_f defined as $K_f = r_fQ_f$. The value $C_f = E_{K_f}(H(ID_i, i*w)G \mod p)$. The composite value C_1 will be defined as,

$$C_1 = \{C_f, A_f\} \tag{7}$$

Similar to fog nodes, the cloud node is registered with the public key $Q_c = d_cG$ while d_c is stored in private. C_2 requires to be decrypted at the cloud node in order to synchronize the counter i and the dynamic identity ID_i.

Aligning to the ECIES, the random value r_{c-1} will be generated in the IoT node. The value $A_{c-1} = r_{c-1}G$ and the symmetric key K_{c-1} will be defined as $K_{c-1} = r_{c-1}Q_c$. The value $C_{c-1} = E_{K_{c-1}}(i, ID_i)$. The value C_2 is defined as,

$$C_2 = \{C_{c-1}, A_{c-1}\} \tag{8}$$

And finally, the message M_1 will be defined as,

$$M_1 = \langle C_1, C_2 \rangle \tag{9}$$

2) Smart contract performs verification of credentials in the blockchain node in the fog layer

The role of the fog node in the proposed architecture is twofold. The fog node operates as the gateway to the IoT nodes to connect to the cloud node over the internet. In addition to that, the node itself operates as a consortium blockchain node that executes the smart contracts on security services. Using the fog node, the key services of the proposed architecture, such as certificate generation can be performed on the fog nodes using the deployed smart contracts. It is assumed that the smart contracts which run on fog nodes have access to the private storage of the fog node locally to obtain the private key of CA.

In the proposed architecture, the identifier of the digital certificate $(H(ID_i, i*w)G$ has been encrypted over the IoT-Fog channel in message M_1. The fog node is able to decrypt C_1 from the ECIES decryption scheme.

From C_1, the fog smart contract generates $K_f = d_fA_f = r_fQ_f$. The smart contract decrypts C_f from the symmetric key K_f and derives $(H(ID_i, i*w)G \mod p)$, which is the identifier of the digital certificate over the blockchain.

The smart contract retrieves the certificate object i, which is represented as $certObj_i$ from the extended storage $(Step[2.a])$.

The smart contract then performs verification on $H(ID_i, i*w)$ value to verify whether the requesting IoT is not malicious and holds the correct private value without obtaining additional knowledge on $H(ID_i, i*w)$. For this process, the EC based ZK non-interactive Schnorr algorithm is used.

The ledger record for the certificate includes $B_i = iG$ and it is required to prove that the ledger record contains i without revealing the knowledge of i to the verifying smart contract, in order to ensure the anonymity and unlinkability. The ledger record for the certificate contains the EC point P_i which has been generated for ZKP. Furthermore the record consists of randomly generated r_i, $A_i = r_iG$, $c_i = H(x_iP_i, r_iP_i, A_i)$ and $s_i = r_i + c_ix_i$.

The smart contract computes $c_i = H(x_iP_i, r_iP_i, A_i)$ from the ledger record and checks whether $s_iG = A_i + c_iB_i$ and $s_iP_i = r_iP_i + c_ix_iP_i$ are valid in order to verify whether the counter is correctly synchronized.

If the verification was successful, the smart contract calculates $primaryHash_i$ such that,

$$primaryHash_i = H(d_f * H(ID_i, i*w)G \mod p) \tag{10}$$

The smart contract checks whether the $primaryHash_i$ is in the ledger to avoid double usage of certificates. If not, the ledger is included with the $primaryHash_i$ into the hash table of the ledger. Since the table contains the irreversible hash values which have been calculated based on the private data, it is ensured that anonymity and unlinkability have not been compromised. If the ledger contains $primaryHash_i$, the execution of the smart contract is terminated with returning $failedDoubleSpendingCheck - 1$.

If the response is valid, the fog node sends the message $M_2 = certObj_i$ towards the IoT node for the certificate generation in $Step[2.b]$. The auxiliary data aux_i retrieved from the $certObj_i$ defined as,

$$aux_i = H(H(ID_i, i*w)G, Cert_i)r_{c-i} + d_c \tag{11}$$

and the derivated private key d_i is defined as,

$$d_i = H(H(ID_i, i*w)G, Cert_i)H(ID_i, i*w) + aux_i \tag{12}$$

Furthermore, the IoT node generates the public key of the key exchange i which is defined as $d_i * G \mod p$. The IoT node also generates the symmetric key $symKey_i$ which will be used in the IoT-CSP channel defined as,

$$symKey_i = H(d_i * d_c * G \mod p) \tag{13}$$

Algorithm 2 Fog blockchain node requesting verification of smart contract

Require: $M_1 = \langle C_1, C_2 \rangle$
$C_1 = \{C_f, A_f\}$
$C_2 = \{C_{c-1}, A_{c-1}\}$
$K_f = d_f A_f = r_f Q_f$
$H(ID_i, i * w)G \mod p = D_{K_f}(C_f)$
$certObj_i = queryExtendedStorage(H(ID_i, i * w * G \mod p))$
$certObj.zkpDataSet = \{s_i, x_i P_i, r_i P_i, A_i\}$
$c_{sc} = H(x_i P_i, r_i P_i, A_i)$
if $(s_i G == (A_i + c_{sc} iG)) \&\& (s_i P_i == (r_i P_i + c_i x_i P_i))$ **then**
  $primaryHash_i = H(d_f * H(ID_i, i * w * G \mod p))$
  if $containsValueInLedger(primaryHash_i)$ **then**
    return $failedDoubleSpendingCheck - 1$
  else
    $addLedgerRecord(primaryHash_i)$
    $M_2 = \langle certObj_i \rangle$
    $M_3 = ackIoTForKey - Cert - Gen(M_2)$
    $M_3 = \langle symKey_i, d_i * G \mod p \rangle$
    $r_{c-2} = generateRandomValue()$
    $A_{c-2} = r_{c-2} G$
    $K_{c-2} = r_{c-2} Q_c$
    $C_{c-2} = E_{K_{c-2}}(primaryHash_i)$
    $C_3 = \{C_{c-2}, A_{c-2}\}$
    $s_f = generateSignature(C_2, C_3)$
    $M_4 = \langle C_2, C_3, s_f \rangle$
  end if
else
  return $failedChallenge$
end if

Including $d_i * G \mod p$ and $symKey_i$, the message M_3 has been defined as,

$$M_3 = \langle d_i * G \mod p, H(symKey_i) \rangle. \tag{14}$$

The message M_3 will be sent to the fog node smart contract in response to the function $ackIoTForKey-Cert-Gen(M_2)$ invocation $Step[2.c]$ in Algorithm 2. The hash value $H(symKey_i)$ will be stored in the ledger in order to ensure the consistency of $symKey_i$ at the CSP.

It is required to encrypt the $primaryHash_i$ value over the Fog-CSP channel. Aligning to the ECIES, the random value r_{c-2} will be generated in the fog node. The value $A_{c-2} = r_{c-2}G$ and the symmetric key K_{c-2} will be defined as $K_{c-2} = r_{c-2}Q_c$. The value $C_{c-2} = E_{K_{c-2}}(primaryHash_i)$. The value C_3 will be defined as,

$$C_3 = \{C_{c-2}, A_{c-2}\} \tag{15}$$

The values C_2 and C_3 will be sent encrypted to the cloud. The messages will be signed, generating s_f via the Schnorr signature scheme. The message M_4 which will be forwarded to the cloud at step $[2.d]$ is denoted as,

$$M_4 = \langle \{C_2, C_3\}, s_f \rangle \tag{16}$$

Algorithm 3 CSP request verification and key establishment smart contract

Require: $M_4 = \langle C_2, C_3, s_f \rangle$
$C_2 = \{C_{c-1}, A_{c-1}\}$
$C_3 = \{C_{c-2}, A_{c-2}\}$
if $s_f == generateSignature(C_2, C_3)$ **then**
  $K_{c-2} = d_c A_{c-2} = r_{c-2} Q_c$
  $primaryHash_i = D_{K_{c-2}}(C_{c-2})$
  if $containsValueInLedger(primaryHash_i)$ **then**
    $secondaryHash_i = H(d_c * H(ID_i, i))$
    if $containsValueInLedger(secondaryHash_i)$ **then**
      return $failedDoubleSpendingCheck - 2$
    else
      $addLedgerRecord(secondaryHash_i)$
      $K_{c-1} = d_c A_{c-1} = r_{c-1} Q_c$
      $(i, ID_i) = D_{K_{c-1}}(C_{c-1})$
      $symKey_i = H(d_i * d_c * G \mod p)$
      if $containsValueInLedger(symKey_i)$ **then**
        $synchronizeCloudWithIoT(i, ID_i)$
      else
        return $symmetricKeyInvalid$
      end if
    end if
  else
    return $invalidPrimaryHash$
  end if
else
  return $failedSignatureVerification$
end if

3) Cloud synchronization and key establishment

Once the certificate has been generated in the IoT end, it is required to be synchronized with the cloud node on the dynamic ID ID_i, counter i and establish the common symmetric key. Algorithm 3 includes the steps executed in the CSP smart contract to synchronize the IoT node.

The CSP receives the message M_4 from the fog node. The CSP verifies the Schnorr signature s_f with the data received. If the signature does not match, a $failedSignatureVerification$ error will be returned and the smart contract will be terminated. If the signature is correct, the CSP extracts $C_2 = \{C_{c-1}, A_{c-1}\}$ and $C_3 = \{C_{c-2}, A_{c-2}\}$. Using the public key $Q_c = d_c G$, CSP defines the corresponding symmetric keys $K_{c-1} = d_c * A_{c-1} = d_c * r_{c-1} * G$ and $K_{c-2} = d_c * A_{c-2} = d_c * r_{c-2} * G$.

The smart contract decrypts C_{c-1} and derives $\langle i, ID_i \rangle$ as,

$$\langle i, ID_i \rangle = D_{K_{c-1}}(i, ID_i) \tag{17}$$

Furthermore, the smart contract decrypts C_{c-2} as,

$$\langle primaryHash_i \rangle = D_{K_{c-2}}(primaryHash_i) \tag{18}$$

The smart contract checks whether the ledger contains $primaryHash_i$ which ensures that the fog verification has been performed previously on the i-th key exchange. If the ledger contains $primaryHash_i$, the smart contract generates the $secondaryHash_i$ such that,

$$secondaryHash_i = H(d_c * H(ID_i, i)) \tag{19}$$

The smart contract checks whether $secondaryHash_i$ is in the ledger to ensure that the certificate has not been used twice. If the ledger contains $secondaryHash_i$, the smart contract terminates the returning $failedDoubleSpendingCheck - 1$. If the $secondaryHash_i$ is not added to the ledger it reflects

that the certificate has not been used previously. Furthermore, the $secondaryHash_i$ record will be added to the ledger. The $symKey_i$ between IoT-CSP channel is defined as,

$$symKey_i = H(d_c * Q_i \mod p) \quad (20)$$

Note that the public key of the IoT device for the session i is stored in the ledger.

This symmetric session key will be used to encrypt the data exchanged between the IoT node and the CSP after the key exchange i. The smart contract checks whether the $symKey_i$ is in the ledger before synchronization of the dynamic ID ID_i and i. If the ledger does not contain the value, the smart contract terminates by returning $symmetricKeyInvalid$ error.

After the successful completion of all these steps, each IoT node has the newly generated dynamic ECQV certificate for the i-th session, public/private keys for each session, and the symmetric keys with the cloud. The ledger contains the pointer of the ECQV certificates, the hash of the public key, and the symmetric key. The CSP has the validated public key of the IoT node, and the symmetric key. The IoT and CSP are now capable in establishing a secure channel between the IoT and CSP using the symmetric key established in the key exchange.

IV. Experimental Evaluation

The experimental setup consists of several Raspberry Pi devices, one host machine, and a few virtual machines. Each VM is installed with Ubuntu 20.04 server. The VMs are deployed with Hyperledger Fabric blockchain service, Message Queuing Telemetry Transport (MQTT) brokering service, and InterPlanetary File System (IPFS) distributed storage service. The host machine processor is an Intel(R) Core i5 -8250 with 32GB RAM. Figure 3 provides an overview of the implementation setup. Algorithm 2 and Algorithm 3 were encoded as Java-based smart contracts in the blockchain platform. Java BouncyCastle library has been used to develop the on-chain cryptographic operations. IPFS library is used to integrate the distributed storage with the smart contract. To simulate the storage of the cloud-based CA ([29]) for comparison, an optimal searchable database ElasticSearch has been used.

We have used the 5G Test Network (5GTN) for the connectivity establishment of the IoT nodes, fog nodes and the cloud. The 5G test network facilitates an industry grade testing and connectivity functions for experimental evaluation. 5GTN provides edge computing resources as well as Giga-bit per second (Gbps) ranged faster connectivity for IoT, fog and cloud layers and eventually simulate industry-grade telecommunication infrastructure. 5GTN also facilitates network softwarization related experiments, such as network slicing, and local 5G operator establishment. In our experiment, 5GTN is used to interconnect the IoT tenant to to the fog later devices which are also located at 5GTN. Finally, we used the the high-speed Internet connection offered by 5GTN backhaul to connect the cloud layer.

The evaluation of the proposed architecture is performed in the three experiments, i.e., 1) Blockchain storage utilization when scaling up IoT nodes, 2) Certificate search latency and 3) End-to-end latency of authentication and key agreement for different block mining intervals. The objectives of experiments include the blockchain storage utilization comparison with the proposed work and state of the art. We programmatically generated thousands of transactions as well as ledger records corresponding to simulate a massive quantity of transaction volume which is similar to an industry-grade traffic volume.

A. Blockchain storage utilization

In the proposed system, we store the dynamic ECQV certificate as IoT node registration entry in the extended storage to reduce the blockchain storage utilization when scaled up. The storage service is deployed in a separate computing node (eg. a virtual machine within the same network) to store the dynamic registration details (Table III) of the IoT node in the extended storage. We evaluated the storage overhead advantage (on-chain stored vs proposed off-chain stored) in the

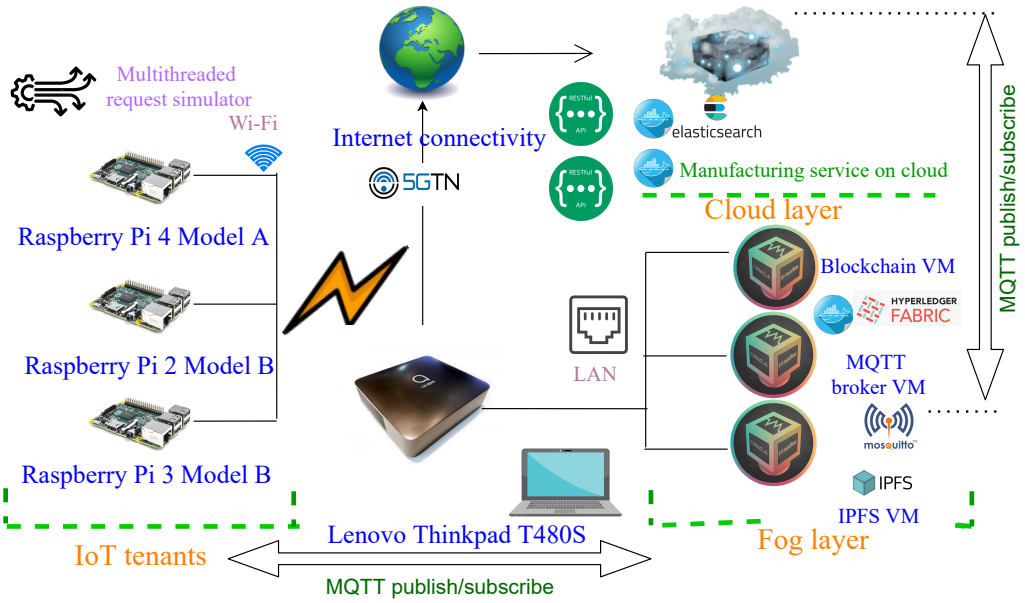


Fig. 3: The implementation setup

proposed architecture. Furthermore, we partially implemented an RSA based architecture on Hyperledger blockchain, relying on the one proposed in [30], [31] for comparison. We selected these related works since both of the work facilitates IoT security services with the integration of blockchain. The use of ECC significantly reduces the key length and we used RSA-3072 bit key to match the similar security level of our proposal. We evaluated the on-chain storage utilization for ECDSA 256 bit and 384 bit dynamic certificates (Table III). In the proposed architecture, the 384 bit dynamic ECDSA certificate object record is represented by Table II. Using the distributed storage pointer $ExtendedStorageKey$ instead of dynamic certificate object reduces the blockchain storage utilization significantly in the proposed architecture. In the implementation, all certificate objects and address pointer are encoded in Base58 form for the consistency.

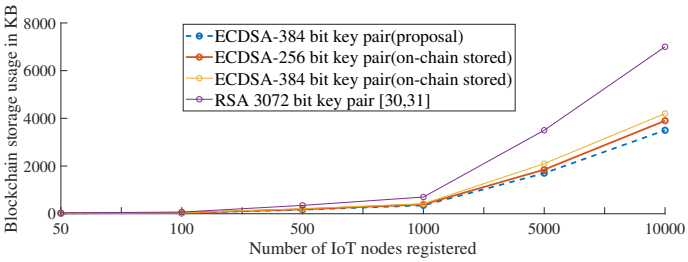


Fig. 4: Blockchain storage utilization for certificates

In the experiment, we used CouchDB storage statistics to monitor the storage utilization of a blockchain node. From the observations, the proposed architecture utilizes minimal on-chain storage when compared with the simulation of [30], [31]. The results are reflected in Figure 4. From the results, we observed that the proposed architecture yields a lower storage utilization when comparing with state of the art. The off-chain storage minimizes the blockchain storage utilization, which will eventually reduce the storage overhead on the fog infrastructure.

B. IoT node search latency

The manufacturing equipment, which is registered as IoT nodes, will be searched on the blockchain for different purposes. For instance, to validate the existence of a dynamic certificate of the IoT node, the ledger storage will be searched to retrieve the certificates. However, the search operation of the proposed architecture incurs additional latency for searching on the IPFS distributed storage, and we evaluated the search latency on the blockchain when the number of dynamic certificates registered has been scaled up. Furthermore, we compared the search latency with a simulated cloud-based Public Key Infrastructure (PKI) service based on [29]. In the simulation, we assumed that the cloud-based PKI service stores the certificate entries in Elastic Search storage, which is efficient in searching. The key reason for using ElasticSearch in the cloud-based simulation is to minimize the latency occurred by storage search. We compared the search latency for 100 trials varying the number of IoT nodes registered in the system. The results are reflected in Figure 5. From the search latency comparison, we observed search latency advantage of the proposal when compared with cloud-based PKI.

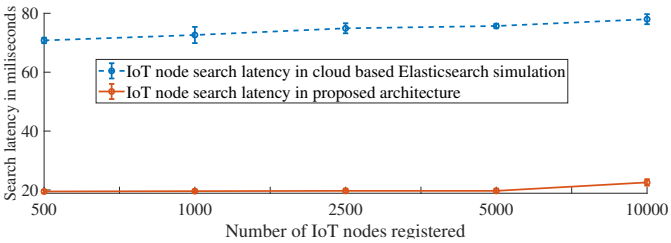


Fig. 5: IoT node search latency comparison

From the experimental evaluation, the proposed work yields better search latency when compared with the cloud-based PKI. Even though the records are stored in extended storage, the search latency is lower since the search operation is performed on the blockchain node itself, instead of searching on a cloud service. In cloud manufacturing applications, the manufacturing IoT nodes which are connected to the fog node will yield lower search latency when compared with searching on the cloud. The results reflect that the proposed architecture supports scaling up the number of registered IoT nodes without a significant impact on the search latency. .

C. Session-based certificate activation and IoT-CSP channel key establishment

In this experiment, we evaluated the end-to-end latency on Step $1-3$ of Figure 2 when gradually increasing the concurrent transaction volume. Furthermore, we modified the proposed architecture with cloud-based MQTT routing to compare additional latency incurred when the proposed work is deployed as a cloud-based security service. In this experiment, the transactions are generated concurrently as batches of 1, 5, 10, 25, and 50 using a multi-threaded software program. For a selected concurrent transaction count, 100 trials are performed to measure the entire transaction batch completion latency. The mean transaction completion time and standard deviation for the entire batch has been recorded. The same set of experiments are performed for different block mining intervals ($BatchTimeout$) configuration in Hyperledger Fabric) $(0.5s, 1s, 2s)$. We simulated the cloud-based message routing in the experiment in order to distinguish the latency advantage of decentralized service architecture. We deployed the MQTT message brokering service in the cloud server, which simulates the cloud transit included round trips in message flow. Figure 6 reflects the results. The evaluated transaction completes when the execution of Algorithm 2 and Algorithm 3 have been completed. At the end of each algorithm, a new block is mined and the end-to-end latency measurements include communication latency and block mining latency.

From the results, we observed that the block mining time is not the sole factor that affects the transaction completion latency. For instance, when the transaction throughput is 25, the lowest block mining time reports the highest latency when compared with $1s$ and $2s$ block mining time configurations. We observed that when the block mining time is lower, the batch of transactions is dispersed across adjoining blocks (The block was cut due to time that has been expired. The remaining transactions of the batch are included in the next block).

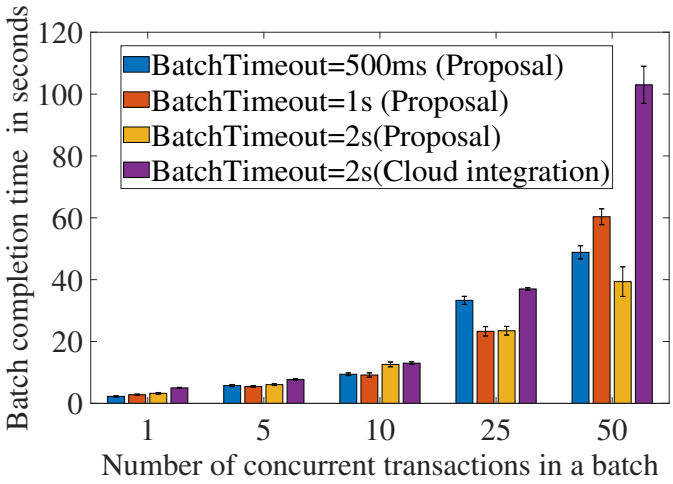


Fig. 6: Certificate activation and IoT-CSP channel key establishment

The entire batch of transactions is complete when the entire set of blocks has been completely mined. However, cloud-based routing indicates the highest end-to-end latency as well as comparably higher standard deviation. Obviously, cloud-based message routing incurs more latency and more variance due to communication overheads. The maximum throughput evaluated in the experimental setup is aligned with production grade throughput since the key exchange is one time operation per session. The key exchange is executed prior to exchanging the manufacturing related transactions.

V. Security Analysis

The security analysis defines the position of the proposed solution in terms of security features mentioned in Section II. In short, the entire proposed architecture utilized well-established and verified algorithms and is relying on ECDLP and ECDHP.

Privacy: The proposed system ensures privacy in the IoT-CSP channel within the manufacturing process. It is assumed that the IoT-CSP channel will be used to exchange manufacturing-related information and a dynamic session key will be established between IoT and CSP using DH key exchange mechanism. The symmetric key is exchanged dynamically at each session when the new ECQV certificate has been generated for the IoT node. Furthermore, the data exchanged within the Fog-Cloud channel for the session key establishment is encrypted using ECIES.

Integrity: The integrity of the key exchange transaction data is ensured in the proposed architecture using the immutable blockchain and integrity preserved distributed storage (implemented in IPFS). The baseline principle to ensure integrity in blockchain is the digital signature. The blockchain stores the address pointers of the ECQV certificates and hash values generated in Algorithm 2 and Algorithm 3. In addition to the digital signatures used in the blockchain, the Schnorr signature scheme has been used to ensure the integrity in messages exchanged between IoT, fog and cloud smart contracts execution.

Authentication: Authentication of the IoT nodes is ensured using the dynamically generated lightweight ECQV certificates. The certificates are accessible from the ledger. The IoT nodes which hold valid certificates will be identified as trusted IoT nodes. To reduce the security risks of the private key compromise of an IoT node, the certificates are generated dynamically for each session. In addition, ZKP ensures that the requested entity possesses the corresponding private key.

Anonymity and unlinkability: Anonymity and unlinkability of the transaction data are ensured using the hashing techniques and non-interactive ZKP in the transaction records. The transaction data which are used in the proposed architecture, including Equation 2, and Table II do not reveal the identity as well as transaction counter i related information in the ledger records. The ledger is completely unaware of the underlying values in the irreversible hash records exchanged in the key certificate activation transactions. Anonymity and unlinkability enable one CSP to facilitate many manufacturing groups even though each of them is a competitor. Each manufacturing group can integrate to the CM system as a consortium member by connecting the fog computing node. The proposed architecture does not reveal individual transaction information on the blockchain.

Replay and re-use prevention: Even though the transaction data preserves anonymity and unlinkability, the data is still verifiable against replay attacks. In the proposed architecture, the session counter i is verified at the Algorithm 2 using non-interactive ZKP. Furthermore, in Algorithm 2, the ledger is checked for existence of $primaryHash_i$ in Algorithm 2 and $secondaryHash_i$ in Algorithm 3.

Forward secrecy: The proposed solution requires the private value w of Equation 2 to be handled and stored privately in the manufacturing equipment owner's end. In terms of forward secrecy, the proposed solution does not provide forward secrecy since the compromise of w allows the external parties to compute the $H(ID_i, i * w)G$ values which will be stored in the blockchain. Thus, it is the manufacturing owner's responsibility to store the private value w in a secured storage in order to ensure the privacy and unlinkability of the transactions over the blockchain.

VI. Discussion and Future Work

Table IV summarizes the limitations of the state of the art and feature-wise comparison with key related works. From the results reflected in the Table IV, the proposed work eliminates the limitations enumerated through the integration of blockchain and extended storage service. In principle, extended storage integration provides flexibility of storage re-usage which is harder to perform in the ledger. For instance, in [32] IPFS storage recycling was proposed. The lightweight dynamic certificates enable more storage scalability to onboard a massive number of consumers when comparing with the state of art. From the results, in numerical and feature-wise perpective, the proposed work outperforms state of art. The proposed fog-based decentralized architecture is possible to re-design with cyber physical manufacturing systems to be connected with cloud using the devices such as mobile phones.

In the industrial evolution perspective, the proposed work facilitates the security of connected infrastructure in the In-

TABLE IV: Limitations of the state of art and comparison of main features with state of the art

Limitations	**Solution in the proposal**	[8]	[9]	[10]	[11]	[18]	Ours
Certificate and public key data structure storage overhead for IoT and the blockchain ledger	Application of lightweight ECQV certificates with smaller data structures for node authentication	N/P	P	P	N/P	N/P	P
Blockchain storage expansion overhead when the number of consumers are being increased.	Scalable blockchain storage with external storage service integration.	N/P	N/P	N/P	N/P	N/P	P
Transaction data linkability over public ledger records	Anonymity and unlinkability on transaction data	N/P	P	N/P	N/P	P	P
Central point of failure which incurs a scalability and latency bottlenecks	Decentralized operation with lower latency and distributed operational capabilities	N/P	N/P	N/P	P	P	P

Note: P represents that a viable solution is proposed by each research work to mitigate the relevant limitations. N/P represents that a viable solution is not proposed by each research work to mitigate the relevant limitations.

dustry 4.0. However, the decentralization, scalability provisions advantages as well as lower latency capabilities of the proposed architecture reflect the security service delivery strengths to comply with human-machine oriented service anticipations in the Industry 5.0 . The proposed architecture designed with the extension provisions towards domains beyond CM. For an instance, Wireless Body Area Network integration for the realtime human health monitoring is one of the interesting contexts in the Industry 5.0. Incorporation of the proposed solution with local 5G operators for realtime health monitoring will provide authentication, anonymity, and faster connectivity for human health monitoring and incident response.

VII. Conclusion

To resolve the trust establishment problem, IoT-Fog-Cloud channel privacy and storage scalability problem, we proposed a blockchain and Fog computing based distributed security services architecture. The proposed architecture provides privacy, integrity, authentication with anonymity and unlinkability of transaction data over the blockchain records. We evaluated the proposed architecture with an experimental implementation setup and compared the results with partially implementing few state of the art architectures. From the results, the proposed architecture outperforms storage overheads, end-to-end transaction latency, and search latency.

References

[1] X. Xu, "From Cloud Computing to Cloud Manufacturing," *Robotics and computer-integrated manufacturing*, vol. 28, no. 1, pp. 75–86, 2012.

[2] "characterization of cyber-physical sensor systems."

[3] Y. Lu and F. Ju, "Smart Manufacturing Systems based on Cyber-physical Manufacturing services (CPMS)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 883–15 889, 2017.

[4] A. Ren, D. Wu, W. Zhang, J. Terpenny, and P. Liu, "Cyber Security in Smart Manufacturing: Survey and Challenges," in *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2017, pp. 716–721.

[5] P. Wang, R. X. Gao, and Z. Fan, "Cloud computing for cloud manufacturing: benefits and limitations," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, 2015.

[6] R. Henzel and G. Herzwurm, "Cloud Manufacturing: A state-of-the-art Survey of Current Issues," *Procedia CIRP*, vol. 72, pp. 947–952, 2018.

[7] S. D. C. Avasalcai, I. Murturi, "Edge and Fog: A Survey, use cases, and Future Challenges ," *Wiley, ISBN 9781119551690, 2020*, 2020.

[8] F. T. Q.Li, "A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing," *IEEE Access, 7, pp 86769 - 86777, 2019*, 2019.

[9] S. Patonico, A. Braeken, and K. Steenhaut, "Identity-based and Anonymous Key Agreement Protocol for Fog Computing Resistant in the Canetti–Krawczyk Security Model," *Wireless Networks*, pp. 1–13, 2019.

[10] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key Management Protocol with Implicit Certificates for IoT Systems," in *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, 2015, pp. 37–42.

[11] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, and M. Guizani, "Blockchain-assisted Secure Device Authentication for Cross-domain Industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[12] A. B. P. Shabisha, K. Steenhaut, "Anonymous Symmetric Key Based Key Agreement Protocol for Fog Computing," *IEEE IoT Journal*.

[13] O. Bouachir, M. Aloqaily, L. Tseng, and A. Boukerche, "Blockchain and Fog Computing for Cyberphysical Systems: The Case of Smart Industry," *Computer*, vol. 53, no. 9, pp. 36–45, 2020.

[14] T. R. Gadekallu, Q.-V. Pham, D. C. Nguyen, P. K. R. Maddikunta, N. Deepa, B. Prabadevi, P. N. Pathirana, J. Zhao, and W.-J. Hwang, "Blockchain for Edge of Things: Applications, Opportunities, and Challenges," *IEEE Internet of Things Journal*, 2021.

[15] U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar, and M. Alazab, "Blockchain for Industry 4.0: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 79 764–79 800, 2020.

[16] T. M. Fernandez-Carames and P. Fraga-Lamas, "A Review on the Application of Blockchain to the Next Generation of cybersecure Industry 4.0 Smart Factories," *IEEE Access*, vol. 7, pp. 45 201–45 218, 2019.

[17] N. Mohamed and J. Al-Jaroodi, "Applying blockchain in industry 4.0 applications," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*. IEEE, 2019, pp. 0852–0858.

[18] F. L. M. Ma, G. Shi, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT ," *IEEE journal of Internet of Things, 14(8), pp. 1184-1195, 2018*, 2018.

[19] Y. Zhang, X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, "Blockchain-based Trust Mechanism for IoT-based Smart Manufacturing System," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1386–1394, 2019.

[20] S. B. ElMamy, H. Mrabet, H. Gharbi, A. Jemai, and D. Trentesaux, "A Survey on the Usage of Blockchain Technology for Cyber-threats in the Context of Industry 4.0," *Sustainability*, vol. 12, no. 21, p. 9179, 2020.

[21] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A Lightweight Scalable Blockchain for IoT Security and Anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[22] T. Kumar, E. Harjula, M. Ejaz, A. Manzoor, P. Porambage, I. Ahmad, M. Liyanage, A. Braeken, and M. Ylianttila, "BlockEdge: Blockchain-edge Framework for Industrial IoT Networks," *IEEE Access*, vol. 8, pp. 154 166–154 185, 2020.

[23] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-Based Reliable and Efficient Certificateless Signature for IIoT Devices," *IEEE Transactions on Industrial Informatics*, 2021.
[24] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Cloud Manufacturing: Challenges, Recent Advances, Open Research Issues, and Future Trends," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9-12, pp. 3613–3639, 2019.
[25] Y. Liu, L. Wang, and X. V. Wang, "Cloud Manufacturing: Latest Advancements and Future Trends," *Procedia Manufacturing*, vol. 25, no. 8, pp. 62–73, 2018.
[26] L. Thames and D. Schaefer, "Software-Defined Cloud Manufacturing for Industry 4.0," *Procedia cirp*, vol. 52, pp. 12–17, 2016.
[27] L. Wang and X. Xu, "Advances and Challenges in Cloud Manufacturing," *Journal of manufacturing science and engineering*, 2015.
[28] "Inter Planetary File System," last accessed 25 September 2021. [Online]. Available: https://ipfs.io/
[29] Z. Yu, Q. Wang, W. Zhang, and H. Dai, "A Cloud Certificate Authority Architecture for Virtual Machines with Trusted Platform Module," in *2015 IEEE 17th International Conference on High Performance Computing and Communications*. IEEE, 2015, pp. 1377–1380.
[30] S. Huh, S. Cho, and S. Kim, "Managing IoT Devices using Blockchain Platform," in *2017 19th international conference on advanced communication technology (ICACT)*. IEEE, 2017, pp. 464–467.
[31] D. Pavithran and K. Shaalan, "Towards Creating Public Key Authentication for IoT Blockchain," in *2019 Sixth HCT Information Technology Trends (ITT)*. IEEE, 2019, pp. 110–114.
[32] J. Rupasena, T. Rewa, K. T. Hemachandra, and M. Liyanage, "Scalable Storage Scheme for Blockchain-Enabled IoT Equipped Food Supply Chains," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2021, pp. 300–305.

Tharaka Hewa is a Doctoral student of NetSEC (Network security, trust and privacy) research group at Center for Wireless Communications, University of Oulu, Finland. Tharaka has obtained his Bachelor in computer science degree in year 2012, from University of Colombo, School of Computing, Sri Lanka. He has completed the Master of Science in information security with a distinction pass from University of Colombo, School of Computing, Sri Lanka in 2016. He worked as a Senior Software Engineer in a leading digital payment systems company in Sri Lanka for 5 years. He has joined Nanyang Technological University, Singapore as a Research Associate in 2017. After two years, he has joined Center for Wireless Communications, University of Oulu. His research interests include Blockchain, IIoT, and 5G network slicing.

An Braeken is full time professor at VUB-INDI. Her interests include lightweight security and privacy protocols for IoT, cloud and fog, blockchain and 5G security. She has developed several lightweight security solutions in the healthcare domain in collaboration with University of Oulu, University College Dublin and University of Oxford. Between 2014–2017, she was Short Term Scientific Mission manager in the COST action AAPELE on Architectures, Algorithms and Platforms for Enhanced Living Environments), involving more than 24 different countries in Europe. In 2020, she co-edited the book on IoT Security-Advances in Authentication, published by Wiley.

Madhusanka Liyanage (Senior Member, IEEE) is an Assistant Professor/Ad Astra Fellow and Director of Graduate Research at the School of Computer Science, University College Dublin, Ireland. He is also acting as a Docent/Adjunct Professor at the Center for Wireless Communications, University of Oulu, Finland, and a Honorary Adjunct Professor at the Department of Electrical and Information Engineering, University of Ruhuna, Sri Lanka. He received his Doctor of Technology degree in communication engineering from the University of Oulu, Oulu, Finland, in 2016. From 2011 to 2012, he worked as a Research Scientist at the I3S Laboratory and Inria, Sophia Antipolis, France. He was also a recipient of the prestigious Marie Skłodowska-Curie Actions Individual Fellowship and Government of Ireland Postdoctoral Fellowship during 2018-2020. During 2015-2018, he has been a Visiting Research Fellow at the CSIRO, Australia, the Infolabs21, Lancaster University, U.K., Computer Science and Engineering, The University of New South Wales, Australia, School of IT, University of Sydney, Australia, LIP6, Sorbonne University, France and Computer Science and Engineering, The University of Oxford, U.K. He is also a senior member of IEEE. In 2020, he received the "2020 IEEE ComSoc Outstanding Young Researcher" award by IEEE ComSoc EMEA. In 2021, he was ranked among the World's Top 2% Scientists (2020) in the List prepared by Elsevier BV, Stanford University, USA. Also, he was awarded an Irish Research Council (IRC) Research Ally Prize as part of the IRC Researcher of the Year 2021 awards for the positive impact he has made as a supervisor. Dr. Liyanage's research interests are 5G/6G, SDN, IoT, Blockchain, MEC, mobile, and virtual network security. More info: www.madhusanka.com

Mika Ylianttila (M. Sc, Dr.Sc, eMBA) (Senior Member, IEEE) is a full-time associate professor (tenure track) at the Centre for Wireless Communications - Networks and Systems research unit, at the Faculty of Information Technology and Electrical Engineering (ITEE), University of Oulu, Finland. He is the head of the NetSEC (Network security, trust and privacy) research group which studies and develops secure, scalable and resource-efficient techniques for 5G and beyond 5G and IoT systems. He has co-authored more than 200 international peer-reviewed articles. He is a Senior Member of IEEE and associate editor in IEEE Transactions on Information Forensics and Security.