AI-Envisioned Blockchain-Enabled Signature-Based Key Management Scheme for Industrial Cyber–Physical Systems

Ashok Kumar Das, *Senior Member, IEEE*, Basudeb Bera, Sourav Saha, *Student Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, Ilsun You, *Senior Member, IEEE*, and Han-Chieh Chao, *Senior Member, IEEE*

***Abstract*—This article proposes a new blockchain-envisioned key management protocol for artificial intelligence (AI)-enabled industrial cyber–physical systems (ICPSs). The designed key management protocol enables key establishment among the Internet of Things (IoT)-enabled smart devices and their respective gateway nodes. The blocks partially constructed with secure data from smart devices by fog servers are provided to cloud servers that are responsible for completing blocks, and then mining those blocks for verification and addition in the blockchain. The most important application of the private blockchain construction is to apply AI algorithms for accurate predictions in Big data analytics. A detailed security analysis along with formal security verification show that the proposed scheme resists various potential attacks in an ICPS environment. Moreover, practical testbed experiments have been conducted using the multiprecision integer and rational arithmetic cryptographic library (MIRACL). Furthermore, a detailed comparative analysis shows superiority of the proposed scheme over recent relevant schemes. In addition, the practical implementation using the blockchain for the proposed scheme demonstrates the total computational costs when the number of transactions per block and also the number of blocks mined in the blockchain are varied.**

***Index Terms*—Authentication, blockchain, industrial cyber–physical systems (ICPSs), key management, security.**

Manuscript received February 10, 2021; revised May 9, 2021 and July 14, 2021; accepted August 27, 2021. Date of publication September 1, 2021; date of current version April 25, 2022. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2020R1I1A2073603, and in part by the Soonchunhyang University Research Fund. *(Corresponding authors: Ilsun You; Han-Chieh Chao.)*

Ashok Kumar Das, Basudeb Bera, and Sourav Saha are with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India (e-mail: iitkgp.akdas@gmail.com; basudeb.bera@research.iiit.ac.in; sourav.saha@research.iiit.ac.in).

Neeraj Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147004, India, also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 413, Taiwan, and also with the School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248 007, India (e-mail: neeraj.kumar@thapar.edu).

Ilsun You is with the Department of Information Security Engineering Soonchunhyang University, Asan 31538, South Korea (e-mail: ilsunu@gmail.com).

Han-Chieh Chao is with the Department of Electrical Engineering, National Dong Hwa University, Hualien City 97401, Taiwan (e-mail: hcchao@gmail.com).

Digital Object Identifier 10.1109/JIOT.2021.3109314

I. Introduction

THE MAIN focus of the cyber–physical systems (CPSs) is to realize the evolution from "traditional and rigid approaches" into "decentralized structures," which aims to implement the "Industry 4.0" as well as the "Industrial Internet visions" [1]. Thus, the deployment of CPS in an industrial scenario becomes very crucial aspect and it needs adequate mechanisms for enabling the high readiness of such systems for the "industrial usage." Several cyber attacks can be mounted by an attacker in the CPS communication layer apart from the threat of failing physical infrastructure. Moreover, in a cloud-based CPS environment, the Internet of Things (IoT)-enabled smart devices, gateway nodes acting as access points, fog, and cloud servers communicate over insecure channels. This poses several potential threats in the Industrial CPS (ICPS) environment, for instance "replay," "Man-in-The-Middle (MiTM)," "impersonation," "privileged insider," "physical smart devices capture," and "ephemeral secret leakage (ESL)" attacks. This suggests design of a robust security protocol in an ICPS environment.

Modern ICPS utilizes the advanced "artificial intelligence (AI)" and "machine learning (ML)" approaches in order to enhance several issues, such as "scalability," "speed" along with the accuracy of ICPS security. In this article, we mainly focus on "accuracy of ICPS security" because of the possibility of "data poisoning attacks." To alleviate such concerns, we design a new blockchain-enabled signature-based key management protocol in an ICPS environment, which will allow the IoT smart devices to securely communicate with their respective gateway nodes. The gateway nodes after forming transactions containing secure data from the smart devices forward those transactions securely to their attached fog servers. Later, the cloud servers in the blockchain center (BC) are in charge of creating, validating, and appending the blocks in the private blockchain.

There are various advantages for using distributed storage against using a centralized system in ICPS. The distributed storage in ICPS can make a system more robust and protect the system from a single point of failure, low latency, and cost effectiveness [2]. In particular, a blockchain-based ICPS system can provide transparency, immutability, and

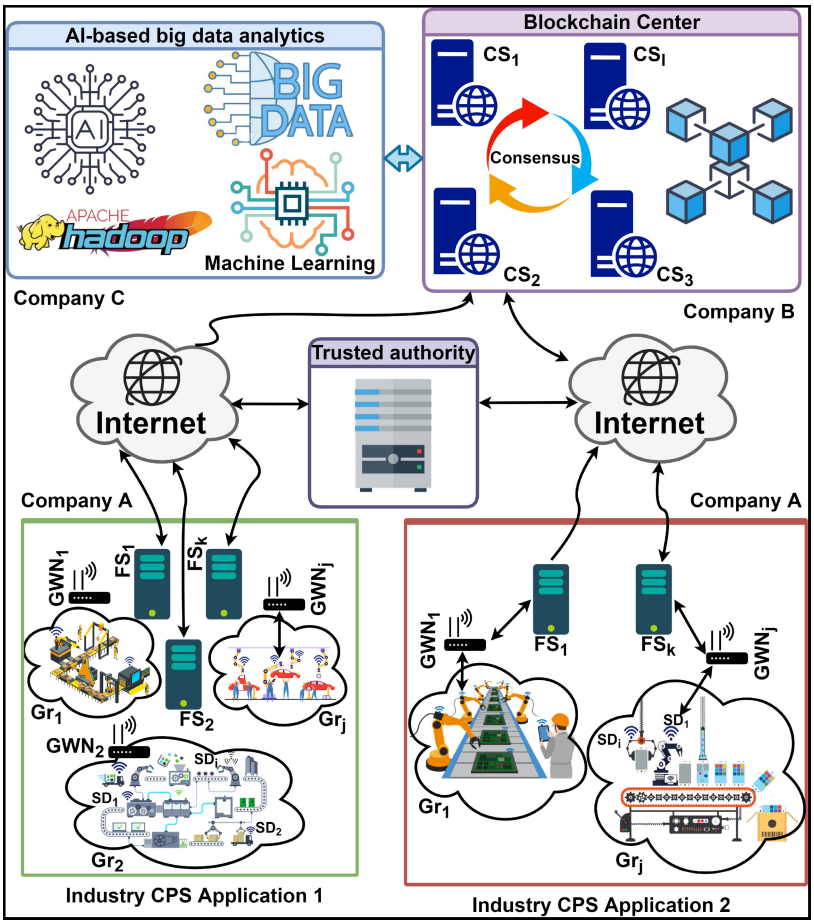


Fig. 1. Blockchain-enabled ICPS.

strong security against various attacks. The ether/gas used in blockchain system refers to as rewards that are given to a miner node for doing the effort or utilizing the computation power by a miner to execute a transaction. The ether/gas is used in the Ethereum-based system which applied Proof-of-Work (PoW) consensus algorithm. To execute a transaction one must need to pay for the computation done by the miner in order to process the transaction [3]. In addition, the gas amount is already defined and there is a drawback if one could not provide enough gas/ether to fulfill the computation, the transaction could be failed. In this work, we need to consider the private blockchain in ICPS context due to the reason that the sensing data from an ICPS application is very sensitive and confidential. Because a private blockchain-based system that we have considered in the ICPS system, it is not encouraged to provide incentives given to a miner. Moreover, the AI/ML-based algorithms are applied on the valid data pertaining in the blocks of the blockchain for correct predictions that will further be very useful for Big data analytics.

A. System Models

The proposed scheme is based on both network and attack models which are discussed below.

1) Network Model: A blockchain-enabled ICPS network model is presented in Fig. 1. Based on each ICPS application belonging to a company, say Company A, we partition the deployed/installed IoT-enabled smart devices into a number of disjoint groups (clusters), say Gr_j $(j = 1, 2, \ldots, n_{gr})$. Each Gr_j will contain a number of IoT smart devices, say SD_i $(i = 1, 2, \ldots, n_{sd})$ and a gateway node GWN_j. All the IoT smart devices SD_i in Gr_j will communicate with their GWN_j securely using the created secret keys through the key management procedure. Each GWN_j in group Gr_j is attached with a fog server, say FS_k $(k = 1, 2, \ldots, n_{fs})$.

The data securely brought by the gateway node GWN_j from its IoT (smart) devices SD_i is used to form the transactions and forward them securely to its FS_k. FS_k will be responsible for creating a partial block from the gathered secure transactions and forwarding the partial blocks to a cloud server CS_l $(l = 1, 2, \ldots, n_{cs})$ for verifying and adding the complete blocks using the "practical Byzantine fault tolerance (PBFT) consensus algorithm" [4] under the Peer-to-Peer (P2P) cloud servers network. The cloud servers belong to another company, say Company B. In addition, we have "AI-based Big data analytic center" belonging to another company, say Company C, where the genuine transactional data stored in private blockchain in the BC are used for appropriate predictions with the help of AI/MI algorithms.

2) Attack Model: We apply the following assumptions and capabilities to an adversary while designing the proposed protocol. First of all, we utilize the broadly applied the "Dolev–Yao (DY) threat model" [5] which assumes that all the communicating entities involved in the network communicate

over insecure (public) channels. An adversary, say $\mathcal{A}$, not only can read the messages in between the communication, but can also alter, remove or place fake information in the communicated messages. Apart from the DY model, under another contemporary *de facto* model, known as "Canetti and Krawczyk's model (the CK-adversary model)" [6], $\mathcal{A}$ can compromise the secret credentials by hijacking session states, and as a result, he/she can also compromise the past or future established secret keys between the communicating parties by means of compromising the session states, session keys and short-term secrets. Thus, the secret keys between the communicating entities need to be constructed using both the long- and short-term secrets to avoid the ESL attack. Furthermore, in general, the IoT smart devices are not treated as trusted nodes in the network. In addition, other entities (gateway nodes, fog servers, and cloud servers) are not also fully trusted [7], [8], and they can be semi-trusted in the network. Therefore, the gateway nodes, fog servers, and cloud servers can be put under locking system as suggested in [9]. However, the IoT smart devices can be physically compromised as they can not be monitored in 24×7. $\mathcal{A}$ can then extract all the secret credentials stored in the compromised IoT smart devices using the power analysis attacks [10]. If the stolen verifier attack is a major potential treat for the gateway nodes, fog servers, and cloud servers, it is assumed that the entities will then store their secret credentials in their respective secure databases or tamper-proof devices so that the secret credentials can not be extracted to $\mathcal{A}$ using the stolen verifier attacks. It is a typical practical assumption. Otherwise, any system is vulnerable to potential attacks when the stolen verifier attack is mounted by $\mathcal{A}$ to extract the secret credentials from the servers.

B. Evaluation Criteria

We discuss the evaluation metrics that will provide a comprehensive measurement to the unsatisfactory "break-fix-break-fix" cycle in the authenticated key management problem.

Wang *et al.* [11] provided a significant step toward breaking the unacceptable break-fix-break-fix cycle in the existing two-factor authenticated key agreement domain. Wang *et al.* [12] provided a set of various criteria that was originally suggested for a "generic client-server architecture." In addition, Wang *et al.* [11] also provided a thorough criteria set containing several individualistic evaluation metrics (criteria) in order to design an authenticated key agreement problem, such as: 1) "no smart card loss attack"; 2) "resistance to various attacks, such as impersonation, offline guessing, replay, MiTM, etc."; 3) "provision of key agreement" for establishing session key between the communicating entities; 4) "sound repairability to support revocation and dynamic sensor node addition phases"; 5) "mutual authentication"; 6) "timely typo detection"; 7) "user anonymity and untraceaility," etc.

C. Research Contributions

The following are the major contributions in this article.

1) A novel blockchain-envisioned signature-based key management protocol for AI-enabled ICPS (BSKMP-ICPS) has been designed. Through the key management procedure, an IoT smart device (SD_i) establishes a secret key with its gateway node (GWN_j) in order to communicate securely between SD_i and GWN_j. Partial blocks constructed by a fog server (FS_k) are forwarded to the BC for verifying and adding the complete blocks using the PBFT consensus algorithm. AI/ML algorithms are then applied on authentic as well as genuine information available in the blocks in the BC to make correct predictions.
2) A thorough security analysis (formal security analysis under the broadly accepted the "Real-Or-Random (ROR) random oracle model" [14], formal security verification using the widely used "automated validation of Internet security protocols and applications (AVISPA) automated simulation software tool" [15], and "nonmathematical security analysis") reveals that BSKMP-ICPS is secure against various attacks that are needed in an ICPS environment.
3) Experiments have been done for various cryptographic primitives in order to show comparative analysis of BSKMP-ICPS using the widely used "multiprecision integer and rational arithmetic cryptographic library (MIRACL)" [16].
4) A thorough comparative analysis shows superiority of the proposed BSKMP-ICPS over "security and functionality features" and comparable "communication and computation costs" as compared to relevant existing schemes.
5) Finally, the blockchain-based implementation of BSKMP-ICPS is done to measure the computational time when the number of transactions per block and the number of blocks mined in the blockchain are varied.

D. Paper Outline

The remainder of this article is sketched as follows. While Section II provides the related work, the detailed phases of the proposed protocol are provided in Section III. The detailed security analysis, including formal security verification is given in both Sections IV and V. The experimental results for various cryptographic primitives using MIRACL are demonstrated in Section VI. A rigorous comparative analysis of the proposed scheme is provided in Section VII. Section VIII gives the blockchain-based implementation of the proposed scheme. Section IX draws the concluding remarks.

II. Related Work

IoT plays an important role in fog computing, wireless multimedia sensor networks, fog-assisted federated learning, prediction-based control plane load reduction in software-defined IoT networks, and so on [17]–[21]. Furthermore, different security related aspects, including security requirements for smart grids, industrial control systems (ICSs), smart cars, and medical devices as parts of ICPS

are discussed by Humayed *et al.* [22]. They also discussed various vulnerabilities and attacks related threats in ICPS. Vegh and Miclea [23], [24] applied the techniques of steganography and cryptography for enhancing the security in ICPS. Choo *et al.* [25] also provided different ICPS-related security innovations as well as challenges.

Sun *et al.* [26] designed a scheme for user authentication and key establishment in a mobile client-server scenario. Their scheme has several security pitfalls, including stolen smart card and replay attacks protection. In addition, their scheme does not support friendly both the password and biometric changes phases [27].

Li *et al.* [13] suggested another scheme for mutual authentication and key establishment using smart card in cloud computing. Their scheme has several security weaknesses, such as protection to stolen smart card, privileged insider, and replay attacks. Moreover, their scheme does not offer friendly password update and biometric change phases [27].

Harishma *et al.* [28] designed an "asymmetric authenticated key-exchange protocol" which targets for both IoT networks and CPS. Their approach is based on "identity-based encryption (IBE)" and the Diffie–Hellman-based key exchange protocol. However, their approach suffers from "the ESL attack under the CK-adversary model and does not support dynamic IoT smart devices addition phase."

Vegh [29] proposed a framework that embeds multifactor authentication for access control along with data analytics in CPS. This scheme does not support dynamic IoT smart devices addition phase. Renuka *et al.* [30] proposed a password-based authentication approach in an IoT-enabled CPS environment that provides four purposes: 1) "a mobile user to mutually authenticate with any gateway node"; 2) "any two IoT smart devices, where at least one of them is connected to a gateway node, to establish a secure connection among each other"; 3) "a mobile user to mutually authenticate with an IoT smart device"; and 4) "any two IoT smart devices that are disconnected from the gateway node to establish a secure connection." Unfortunately, their scheme fails to achieve both anonymity and untraceability properties.

Genge *et al.* [31] designed a "security-aware control application architecture." They also designed a "key distribution protocol" that mainly tailored to address the computational limitations of ICPS. Their approach relies on the "encrypt-then-MAC" authenticated encryption approach. Challa *et al.* [32] designed two authentication protocols in a cloud-based CPS environment, namely: 1) "authentication between a user and a cloud server" and 2) "authentication between an IoT-based smart meter and a cloud server." However, their protocols do not support "anonymity" and "untraceability."

Lara *et al.* [33] also suggested an authentication scheme for the resource-constrained industrial IoT devices. Although their scheme is lightweight, it do not support anonymity and untraceability properties. Furthermore, their scheme does not also support dynamic IoT smart device addition phase after initial deployment. In addition, Jiang *et al.* [34] designed a cloud-centric three-factor authentication and key agreement scheme which integrates passwords, smart cards, and biometrics at the same time to assure secure access to both cloud and autonomous vehicles. Recently, the blockchain technology has been applied for various applications in order to secure the systems [35]–[43].

Finally, various cryptographic techniques, advantages, and disadvantages/limitations of the existing competing authentication/access control schemes in IoT/CPS environment are briefed in Table I.

III. Proposed Protocol

This section discusses the proposed BSKMP-ICPS. For replay attack protection, the current system timestamps are embedded in the communicating messages among the entities during the key management phase. This assumption is practical which is also used in authenticated key management protocols [32], [44]. Table II lists various symbols and their meanings that are used in BSKMP-ICPS. Various phases related to BSKMP-ICPS are discussed in the following.

A. Setup Phase

The *TA* being the trusted entity in the network is responsible for selecting a "nonsingular elliptic curve $E_q(\alpha, \beta)$ of the form: $y^2 = x^3 + \alpha x + \beta \pmod q$" over a Galois (finite) field $GF(q)$ with constants $\alpha, \beta \in Z_q = \{0, 1, \ldots, q-1\}$ having $4\alpha^3 + 27\beta^2 \neq 0 \pmod q$, where q is sufficiently large prime number so that the "elliptic curve discrete logarithm problem (ECDLP)" turns out to be intractable. The *TA* then picks a "base point $G \in E_q(\alpha, \beta)$ whose order is as large as q," "ECDSA signature generation and verification algorithms" [45], and also a "collision-resistant cryptographic one-way hash function" (for instance, we use Secure Hash Algorithms (SHA-256) as a hash function).

Next, the *TA* picks its own private-public keys pair (pr_{TA}, Pub_{TA}), where $pr_{TA} \in Z_q^* = \{1, 2, \ldots, q-1\}$ is randomly generated and $Pub_{TA} = pr_{TA} \cdot G$. It is assumed that all the fog servers FS_k ($k = 1, 2, \ldots, n_{fs}$) and cloud servers CS_l ($l = 1, 2, \ldots, n_{cs}$) are registered by the *TA*. After that each FS_k will generate its own private-public keys pair ($pr_{FS_k} \in Z_q^*$, $Pub_{FS_k} = pr_{FS_k} \cdot G$), and also each CS_l will generate its own private-public keys pair ($pr_{CS_l} \in Z_q^*$, $Pub_{CS_l} = pr_{CS_l} \cdot G$). FS_k and CS_l declare their public keys Pub_{FS_k} and Pub_{CS_l} as public, whereas the *TA* also declares its own public key Pub_{TA} as public. However, the private key of each entity is kept secret to that entity only.

B. Enrollment Phase

The *TA* is responsible for enrolling each IoT smart device SD_i and the gateway node GWN_j for every ICPS application as discussed in the network model (see Section I-A1).

1) Gateway Node Enrollment: For each gateway node GWN_j under a group Gr_j, the *TA* does the following.

Step GNE1: The *TA* first selects a unique identity ID_{GWN_j} and its corresponding pseudo identity as $RID_{\text{GWN}_j} = h(ID_{\text{GWN}_j} || pr_{TA})$ and a temporal credential as $TC_{\text{GWN}_j} = h(ID_{\text{GWN}_j} || RTS_{\text{GWN}_j} || pr_{TA})$, where RTS_{GWN_j} is the GWN_j's

TABLE I
CRYPTOGRAPHIC TECHNIQUES, ADVANTAGES, AND LIMITATIONS OF EXISTING AUTHENTICATION/ACCESS CONTROL SCHEMES

Scheme	Cryptographic Techniques	Advantages	Drawbacks/Limitations
Sun *et al.* (2013)	* Elliptic curve cryptography (ECC) * One-way hash functions	* Mutual authentication * Session key establishment	* Vulnerable to stolen smart card and replay attacks * Does not support friendly both the password and biometric changes phases
Li *et al.* [13] (2015)	* ECC * One-way hash functions	* Mutual authentication * Session key establishment	* Vulnerable to stolen smart card, privileged-insider and replay attacks * Does not support friendly both the password and biometric changes phases
Harishma *et al.* (2018)	* Modular exponentiations * One-way hash functions * Identity-based encryption * Symmetric key encryption/ decryption	* Mutual authentication * Session key establishment	* Vulnerable to ephemeral secret leakage (ESL) attack under CK-adversary model * Does not support dynamic IoT smart devices addition Does not support blockchain security solution
Renuka *et al.* (2019)	* Symmetric key encryption/ decryption * One-way hash functions	* Mutual authentication * Session key establishment	* Does not support anonymity and untraceability * Does not support blockchain solution
Genge *et al.* (2019)	* Message authentication codes (MAC) * Modular exponentiations * Hash functions	* Mutual authentication * Group key distribution	* Does not support blockchain solution
Challa *et al.* (2020)	* ECC * Fuzzy extractor * One-way hash functions	* Mutual authentication * Session key agreement	* Does not support anonymity and untraceability * Does not support security blockchain solution
Lara *et al.* (2020)	* One-way hash functions	* Mutual authentication * Session key agreement	* Does not support anonymity and untraceability * Does not support dynamic IoT smart devices addition * Does not support blockchain solution
Proposed (BSKMP-ICPS)	* ECC * One-way hash functions	* Mutual authentication * Session key agreement * Support blockchain solution	* Needs to implement in real-world environment as future research work

registration timestamp. Next, the *TA* generates a unique "t-degree bivariate symmetric polynomial of the form: $\text{Poly}_j(x, y) = \sum_{u=0}^{t} \sum_{v=0}^{t} a_{u,v} x^u y^v \pmod q$" where the coefficients $a_{u,v} \in Z_q$ are chosen randomly, $\text{Poly}_j(x, y) = \text{Poly}_j(y, x)$ and $t >> n_{sd}$ in each Gr_j so that the proposed key management phase in Section III-C becomes "unconditionally secure and t-collision resistant" against IoT smart device physical capture attacks [46]. Next, the *TA* calculates the polynomial share $\text{Poly}_j(RID_{\text{GWN}_j}, y)$ and sends the credentials $\{RID_{\text{GWN}_j}, TC_{\text{GWN}_j}, \text{Poly}_j(RID_{\text{GWN}_j}, y), h(\cdot), E_q(\alpha, \beta), G\}$ securely to each GWN_j.

Step GNE2: After receiving the credentials from the *TA*, each GWN_j generates its own private-public keys pair ($pr_{\text{GWN}_j} \in Z_q^*$, $Pub_{\text{GWN}_j} = pr_{\text{GWN}_j} \cdot G$), saves pr_{GWN_j} and makes Pub_{GWN_j} as public. The *TA* also deletes the generated credentials ID_{GWN_j}, RID_{GWN_j}, and TC_{GWN_j} in order to thwart against stolen verifier attack. Thus, the GWN stores $\{RID_{\text{GWN}_j}, TC_{\text{GWN}_j}, \text{Poly}_j(RID_{\text{GWN}_j}, y), h(\cdot), E_q(\alpha, \beta), G, pr_{\text{GWN}_j}\}$ in its secure database so that the stolen-verifier attack is prevented to launch other attacks including impersonation attacks.

2) IoT Smart Device Enrollment: For each smart device SD_i in each group Gr_j, the *TA* executes the following steps.

Step SDE1: For each SD_i, the *TA* picks a unique identity ID_{SD_i}, a unique random temporary identity TID_{SD_i}, a registration timestamp RTS_{SD_i} and a private-public keys pair ($pr_{SD_i} \in Z_q^*$, $Pub_{SD_i} = pr_{SD_i} \cdot G$), and computes the temporal credential as $TC_{SD_i} = h(ID_{SD_i} || RTS_{SD_i} || pr_{TA})$ and pseudo identity $RID_{SD_i} = h(ID_{SD_i} || pr_{TA})$, where RTS_{SD_i} is the SD_i's registration timestamp.

TABLE II
NOTATIONS AND THEIR MEANINGS

Symbol	Meaning
TA	Trusted authority
SD_i, ID_{SD_i}	i^{th} IoT smart device and its identity
GWN_j, ID_{GWN_j}	j^{th} gateway node and its identity
TID_{SD_i}	Temporary identity of each SD_i
RID_X	Pseudo identity of an entity X
FS_k, ID_{FS_k}	k^{th} fog server and its identity
CS_l, ID_{CS_l}	l^{th} cloud server and its identity
(pr_X, Pub_X)	Private-public keys pair of an entity X
$h(\cdot)$	A collision-resistant cryptographic one-way hash function
$E(\cdot)/D(\cdot)$	Public key encryption/decryption function
q	A large prime number
$GF(q)$	Galois (finite) field over large prime q
$E_q(\alpha, \beta)$	A non-singular elliptic curve: $y^2 = x^3 + \alpha x + \beta \pmod q$ over $GF(q)$ with $\alpha, \beta \in Z_q = \{0, 1, \cdots, q-1\}$
G	A base point G in $E_q(\alpha, \beta)$
$P + Q$	Elliptic curve point addition of two points $P, Q \in E_q(\alpha, \beta)$,
$k \cdot G$	Elliptic curve point multiplication; $k \cdot G = G + G + \cdots G$ (k times), $G \in E_q(\alpha, \beta)$, $k \in Z_q^*$
ECDSA	Elliptic curve digital signature algorithm
TS_x	Current system timestamp generated by an entity X
ΔT	Maximum transmission delay associated with a message
$x * y$	Modular multiplication of elements $x, y \in Z_q$
$\|\|, \oplus$	Data concatenation & exclusive-OR operators, respectively

Step SDE2: The *TA* also calculates the polynomial share $\text{Poly}_j(RID_{SD_i}, y)$ for each SD_i corresponding to Gr_j and preloads the credentials $\{(TID_{SD_i}, RID_{SD_i}), RID_{\text{GWN}_j}, TC_{SD_i}, \text{Poly}_j(RID_{SD_i}, y), (pr_{SD_i}, Pub_{SD_i}), h(\cdot), E_q(\alpha, \beta), G\}$ into the memory of SD_i prior to its deployment in Gr_j.

Step SDE3: The *TA* sends securely $\{(TID_{SD_i}, RID_{SD_i}) | i = 1, 2, \ldots, n_{sd}\}$ to GWN_j residing in its group Gr_j. The *TA* also deletes the information ID_{SD_i}, TID_{SD_i}, and TC_{SD_i}

Gateway node (GWN_j)
$\{RID_{GWN_j}, TC_{GWN_j}, Poly_j(RID_{GWN_j}, y), pr_{GWN_j}, h(\cdot), \{(TID_{SD_i}, RID_{SD_i}) \mid i = 1, 2, \cdots, n_{sd}\}, E_q(\alpha, \beta), G\}$
IoT smart device (SD_i)
$\{(TID_{SD_i}, RID_{SD_i}), RID_{GWN_j}, TC_{SD_i}, Poly_j(RID_{SD_i}, y), pr_{SD_i}, h(\cdot), E_q(\alpha, \beta), G\}$

Fig. 2. Information preloaded into each GWN_j and SD_i.

to thwart against stolen verifier attack. Furthermore, the *TA* makes Pub_{SD_i} as public for each SD_i.

Various credentials stored in different entities have been listed in Fig. 2.

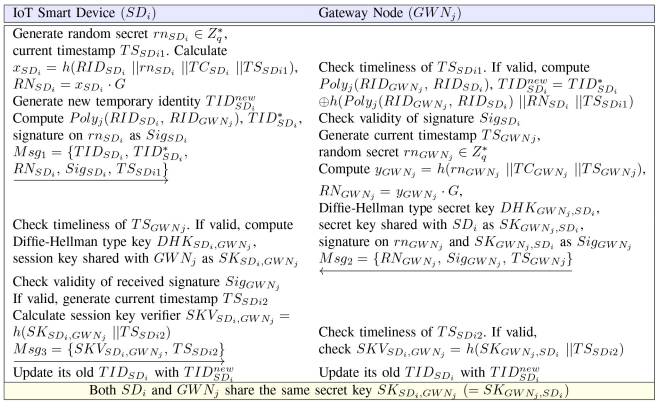


Fig. 3. Overview of key management phase.

C. Key Management Phase

This phase permits a registered IoT smart device (SD_i) to establish a secret key with its associated gateway node GWN_j residing in a group Gr_j for a particular ICPS application. The following are the steps needed for key management between SD_i and GWN_j.

Step KM1: SD_i as the initiator creates a random secret $rn_{SD_i} \in Z_q^*$ and a current timestamp TS_{SDi1} to calculate $x_{SD_i} = h(RID_{SD_i} \| rn_{SD_i} \| TC_{SD_i} \| TS_{SDi1})$ and $RN_{SD_i} = x_{SD_i} \cdot G$. Next, SD_i generates a new temporary identity $TID_{SD_i}^{\text{new}}$, evaluates its own polynomial share $\text{Poly}_j(RID_{SD_i}, y)$ at $y = RID_{\text{GWN}_j}$ to have the secret value $\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j})$, and computes $TID_{SD_i}^* = TID_{SD_i}^{\text{new}} \oplus h(\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j}) \| RN_{SD_i} \| TS_{SDi1})$ and signature on rn_{SD_i} as $Sig_{SD_i} = x_{SD_i} + h(TID_{SD_i} \| TID_{SD_i}^{\text{new}} \| RN_{SD_i} \| Pub_{SD_i} \| TS_{SDi1}) * pr_{SD_i} \pmod q$. After that SD_i sends a key management request message $Msg_1 = \{TID_{SD_i}, TID_{SD_i}^*, RN_{SD_i}, Sig_{SD_i}, TS_{SDi1}\}$ to GWN_j via public channel.

Step KM2: After receiving the message Msg_1 at time TS_{SDi1}^*, the timeliness of TS_{SDi1} is checked by GWN_j with the verifying condition: $|TS_{SDi1} - TS_{SDi1}^*| < \Delta T$, where the significance of ΔT is the "maximum transmission delay." If it is valid, GWN_j fetches RID_{SD_i} of SD_i corresponding to TID_{SD_i} from its secure database. GWN_j then calculates $\text{Poly}_j(RID_{\text{GWN}_j}, RID_{SD_i})$ using its own RID_{GWN_j}, which is essential same as $\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j})$ because the polynomial $\text{Poly}_j(x, y)$ is symmetric, and $TID_{SD_i}^{\text{new}} = TID_{SD_i}^* \oplus h(\text{Poly}_j(RID_{\text{GWN}_j}, RID_{SD_i}) \| RN_{SD_i} \| TS_{SDi1})$. Next, GWN_j verifies the signature Sig_{SD_i} by the condition: $Sig_{SD_i} \cdot G = RN_{SD_i} + h(TID_{SD_i} \| TID_{SD_i}^{\text{new}} \| RN_{SD_i} \| Pub_{SD_i} \| TS_{SDi1}) \cdot Pub_{SD_i}$. If the signature is valid, the next step is executed; otherwise, the phase is immediately terminated by GWN_j.

Step KM3: GWN_j creates current timestamp TS_{GWNj} along with random secret $rn_{\text{GWN}_j} \in Z_q^*$ in order to compute $y_{\text{GWN}_j} = h(rn_{\text{GWN}_j} \| TC_{\text{GWN}_j} \| TS_{GWNj})$, $RN_{\text{GWN}_j} = y_{\text{GWN}_j} \cdot G$, the Diffie–Hellman type secret key $DHK_{\text{GWN}_j,SD_i} = y_{\text{GWN}_j} \cdot RN_{SD_i}$, secret key shared with SD_i as $SK_{\text{GWN}_j,SD_i} = h(\text{Poly}_j(RID_{\text{GWN}_j}, RID_{SD_i}) \| DHK_{\text{GWN}_j,SD_i} \| Sig_{SD_i})$ and also signature on rn_{GWN_j} and SK_{GWN_j,SD_i} as $Sig_{\text{GWN}_j} = y_{\text{GWN}_j} + h(RID_{\text{GWN}_j} \| Pub_{\text{GWN}_j} \| TS_{GWNj} \| SK_{\text{GWN}_j,SD_i}) * pr_{\text{GWN}_j} \pmod q$. After these calculations, GWN_j sends the key management response message $Msg_2 = \{RN_{\text{GWN}_j}, Sig_{\text{GWN}_j}, TS_{GWNj}\}$ to SD_i via open channel. It is worth noticing that we need not to include GWN_j's pseudo identity RID_{GWN_j} because GWN_j has prior knowledge about its all IoT smart devices deployed in Gr_j.

Step KM4: After reception of Msg_2 at time TS_{GWNj}^*, SD_i checks the timeliness of the TS_{GWNj} by the verifying condition: $|TS_{GWNj} - TS_{GWNj}^*| < \Delta T$. If it passes, SD_i will proceed to calculate the Diffie–Hellman type key $DHK_{SD_i,\text{GWN}_j} = x_{SD_i} \cdot RN_{\text{GWN}_j}$, which is same as $DHK_{\text{GWN}_j,SD_i} = y_{\text{GWN}_j} \cdot RN_{SD_i}$, that is, $DHK_{SD_i,\text{GWN}_j} = DHK_{\text{GWN}_j,SD_i}$ and session key shared with GWN_j as $SK_{SD_i,\text{GWN}_j} = h(\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j}) \| DHK_{SD_i,\text{GWN}_j} \| Sig_{SD_i})$. Next, SD_i verifies the validity of received signature Sig_{GWN_j} as $Sig_{\text{GWN}_j} \cdot G = RN_{\text{GWN}_j} + h(RID_{\text{GWN}_j} \| Pub_{\text{GWN}_j} \| TS_{GWNj} \| SK_{SD_i,\text{GWN}_j}) \cdot Pub_{\text{GWN}_j}$. If the signature is valid, SD_i authenticates GWN_j. At the same time, computed session key $SK_{SD_i,\text{GWN}_j} (= SK_{\text{GWN}_j,SD_i})$ is considered as valid one and SD_i proceeds for the next step.

Step KM5: SD_i generates a current timestamp TS_{SDi2} and calculates the session key verifier $SKV_{SD_i,\text{GWN}_j} = h(SK_{SD_i,\text{GWN}_j} \| TS_{SDi2})$. SD_i then sends the final message as the acknowledgment $Msg_3 = \{SKV_{SD_i,\text{GWN}_j}, TS_{SDi2}\}$ to GWN_j via open channel, and updates its old TID_{SD_i} with $TID_{SD_i}^{\text{new}}$ in its memory, which was calculated in step KM1.

Step KM6: After receiving the message Msg_3 at time TS_{SDi2}^*, the timeliness of TS_{SDi2} is checked by GWN_j with the verifying condition: $|TS_{SDi2} - TS_{SDi2}^*| < \Delta T$. If the timestamp is valid, GWN_j checks the verifying condition: $SKV_{SD_i,\text{GWN}_j} = h(SK_{\text{GWN}_j,SD_i} \| TS_{SDi2})$ with the help of its previously calculated session key SK_{GWN_j,SD_i}. If the condition is satisfied, GWN_j authenticates SD_i as valid entity and also considered SK_{GWN_j,SD_i} as authentic. GWN_j updates the old TID_{SD_i} of SD_i with new calculated $TID_{SD_i}^{\text{new}}$ in step KM2 in its secure database.

Finally, the overall key management phase is briefed in Fig. 3.

D. Incorporating Key Management in Block Addition and Verification Into Blockchain

In this section, we discuss how the proposed key management helps in secure data collection for blocks formation in the private blockchain. In the ICPS applications, the sensing data from various IoT smart devices are typically private and confidential to the industries. Hence, the sensitive data should not be leaked in public. In addition, the blockchain application

Block Header	
Block Version	$VerBlk$
Previous Block Hash	$PHBlk$
Merkle Tree Root	MTR_{Blk}
Application Type	$AppType$
Timestamp	TS_{Blk}
Owner of Block	ID_{FS_k}
Public Key of Owner	Pub_{FS_k}
Block Payload (Encrypted Transactions)	
List of n_t Encrypted Transactions $\#w$ (Tx_w)	$\{E_{Pub_{FS_k}}(Tx_w)\|w = 1, 2, \cdots, n_t\}$
ECDSA Signature	Sig_{Blk}
Current Block Hash	$CHBlk$

Fig. 4. Full block *FullBlk* for various transactions.

makes the data secure so that "immutability," "transparency," and "decentralization" properties are preserved.

The creation and operation mechanisms of the blockchain are done using the following steps.

Step 1: First of all, the data coming securely from IoT smart devices SD_i to their respective gateway nodes GWN_j in a group Gr_j are formed into various transactions Tx_w, where Tx_w = {sensing data, identity RID_{SD_i} of smart device SD_i, timestamp TS_{Tx_w}}. These transactions are then securely forwarded to the attached fog server FS_k of GWN_j encrypted using the ECC-based encryption with the public key Pub_{FS_k}.

Step 2: FS_k collects and encrypts the transactions of the form $ETx_w = E_{Pub_{FS_k}}[Tx_w]$, and forms a partial block *ParBlk* on n_t encrypted transactions ETx_w. The Merkle tree root (MTR_{Blk}) is calculated on n_t encrypted transactions ETx_w and also the block creation timestamp (TS_{Blk}), application type of CPS (*AppType*), owner of block (ID_{FS_k}), public key of owner (Pub_{FS_k}), and also the ECDSA signature Sig_{Blk} on the message $m = h(TS_{Blk} \,||MTR_{Blk} \,||AppType \,||ID_{FS_k} \,||Pub_{FS_k} \,||$ encrypted transactions (ETx_w) $(1 \leq w \leq n_t))$ using the FS_k's private key pr_{FS_k} are calculated.

Step 3: Next, each FS_k forwards the formed partial blocks to their associated cloud servers CS_l, which are encrypted with the CS_l's public key Pub_{CS_l}. The in-charge CS_l adds the "block version (*VerBlk*)," "previous hash block (*PHBlk*)," and "current hash block (*CHBlk*) on the data containing *PHBlk* and *ParBlk*." Such a full block *FullBlk* is shown in Fig. 4.

Step 4: A full block *FullBlk* is finally checked and inserted into the private blockchain of the BC using the "PBFT consensus algorithm" [4] under the P2P CS network. This job is explained in Algorithm 1.

Remark 1: Based on our network model provided in Fig. 1, there are multiple companies in this system working together (e.g., all the IoT smart devices and fog servers are under a company, say company *A*; the cloud servers belong to a company, say company *B*; and the AI-enabled Big data analytics under another company, say company *C*). It is worth noticing that we have applied the voting-based PBFT consensus mechanism in order to verify and add a full block in the P2P cloud servers network. As a result, we have not considered the incentives for the fog servers and cloud servers to work as miners and building new blocks in the blockchain. Furthermore, it is also noted that a partial block *ParBlk* contains the encrypted transactions $ETx_w = E_{Pub_{FS_k}}[Tx_w]$, the Merkle tree root MTR_{Blk} and also the ECDSA signature Sig_{Blk}, and *ParBlk* is then forwarded to its associated cloud server CS_l. CS_l then verifies the signature and it is valid, the partial block *ParBlk* is also verified using by checking if the calculated Merkle tree root on the encrypted transactions ETx_w matches with the stored MTR_{Blk} in the block. If it is valid, CS_l ensures the integrity of the partial block *ParBlk*. Furthermore, the encrypted transactions ETx_w and signature Sig_{Blk} can not computed by any cloud servers including CS_l and an adversary, because it needs a fog server FS_k's private key pr_{FS_k}. Later, CS_l adds the previous block hash and current block hash with the partial block *ParBlk* to convert it into a full block *FullBlk* (see Fig. 4). Hence, the proposed scheme handles security and privacy issues of data outsourcing.

Algorithm 1 Consensus for Block Verification and Addition

Input: A full block *FullBlk*; private-public keys pairs $\{(pr_{CS_l}, Pub_{CS_l})\}$, and $n_{f_{CS_l}}$: the number of faulty nodes in P2P network.
Output: Commitment and addition of *FullBlk*.

1: A leader, say L is chosen by a cloud server CS_l based on round-robin policy.
2: L generates timestamp TS_L and sends a request message $\langle FullBlk, E_{Pub_{CS_l}}[V_{req}, TS_L], TS_L\rangle$ to each other peer cloud server CS_l via public channel, where V_{req} is the voting request.
3: After receiving request message, each CS_l checks timeliness of TS_L. If it is valid, CS_l computes $(V_{req}, TS'_L) = D_{pr_{CS_l}}[E_{Pub_{CS_l}}[V_{req}, TS_L]]$, and then verifies TS'_L with the received TS_l.
4: If the validation is successful, CS_l also verifies MTR_{Blk}, Sig_{Blk} and *CHBlk* on *FullBlk*.
5: After all the validations pass successfully, CS_l sends its response message $\langle E_{Pub_L}[V_{res}, TS_{CS_l}], TS_{CS_l}\rangle$ to L via public channel, where V_{res} is the voting response and TS_{CS_l} is current timestamp.
6: Let V_c signify the number of valid votes. Set $V_c \leftarrow 0$.
7: **for** each response message $\langle E_{Pub_L}[V_{res}, TS_{CS_l}], TS_{CS_l}\rangle$ from other cloud servers CS_l **do**
8: L checks timeliness of timestamp TS_{CS_l}, decrypts message using its private key pr_L and validates TS_{CS_l} and V_{res}. If all these terms are valid, set $V_c = V_c + 1$.
9: **end for**
10: **if** $(V_c > 2n_{f_{CS_l}} + 1)$ **then**
11: L transmits commit response for successful verification of *FullBlk* to its all followers CS_l.
12: Each CS_l including L add *FullBlk* to their respective ledgers.
13: **end if**

E. AI-Based Secure Prediction Using Blockchain

Based on a report provided in [47], in 2018, about 2.8 billion consumer data records were leaked in 342 breaches, ranging from credential stuffing to ransomware, with an approximated cost of more than U.S. $654 billion. In 2019, it has further expanded to an exposure of 4.1 billion data records. Till date, the utilization of AI and ML as a prime offensive method in cyber attacks field has not become as a mainstream. However, its application and abilities are also enlarging and enhancing in more sophisticated manner. It is expected that the cyber

criminals will make use of the advantages of AI/ML. Thus, such kind of move will certainly rise various security threats to digital security, and also at the same time, it will increase the volume and filtering of cyber attacks [47].

In general, if we have larger data sets, they will help in creating better ML models. At the same time, the quality of the data is also equally important. The data sets require to be upgraded with recent and relevant data (information) to make the models effective [48]. It is worth noticing that the data is primary concern to AI effectiveness, and the blockchain technology further enables collective and secure data sharing. Thus, the blockchain will certainly assure the trustworthiness of data, and it can further enable more data to be securely shared prior to AI extracts insights from it [48].

The transactional data stored in the blockchian are of heterogeneous types having image file and text file. For the Big-data analytics based on AI/ML, the transactional data in the blockchain will play an important role in AI/ML performance. It is worth noticing that the ML makes utilization of two kinds of data sets: 1) "training data set" and 2) "testing data set," which are divided from a unique Big data set. Next, by utilizing various ML algorithms, such as "linear regression," "logistic regression," and "support vector machine (SVM)," the prediction outputs (classifier) are accordingly created for the future game plan. Now, the problem arises when the data set comes as poisonous, and as a result, it may lead to incorrect prediction results.

A poisoning attack may occur if an adversary injects malicious data into a model's training pool. An attacker can insert false data, modify the label of data, and remove data or insert random noise to the data to poison the training data. In a model attack, an attacker may pollute a model's hyper-parameters that are learned using AI/ML algorithms [49]. Thus, the model performance is dependent on the hyper-parameters. As a result, modification of the hyper-parameters can have significant drop in the model performance. In case of model stealing attacks, an attacker can steal or duplicate models or recover training data membership via black-box probing [50]. Another attack, known as data poisoning attack, is considered as a potential threat in AI/ML, because fake data inserted by malicious users may help in deluding the "training data sets for puzzling AI/ML algorithms" [44], [51], and as a result, it will subsequently lead to "incorrect predication on data sets." Some examples of data poisoning attacks include the Poisson noise insertion and label flipping attacks. Such attacks become serious as they may play a significant factor for businesses and organizations for both "financial terms" and "damaging their reputations." Apart from the data poisoning attack, other attacks, such as "MiTM and impersonation attacks" can be launched as cyber attacks. However, using the proposed blockchain-based key management protocol, the transactional data stored in private blockchain are authentic and genuine. This helps in avoiding "data poisoning attacks and other attacks by an adversary, and it leads to run the AI/ML algorithms as per expectations in order to produce the correct predictions for the Big data analytics purpose." This is mainly possible as the blocks are verified in the blockchain before taking then into account for "data analytics purpose on the decrypted transactions" by the fog servers depending on application types.

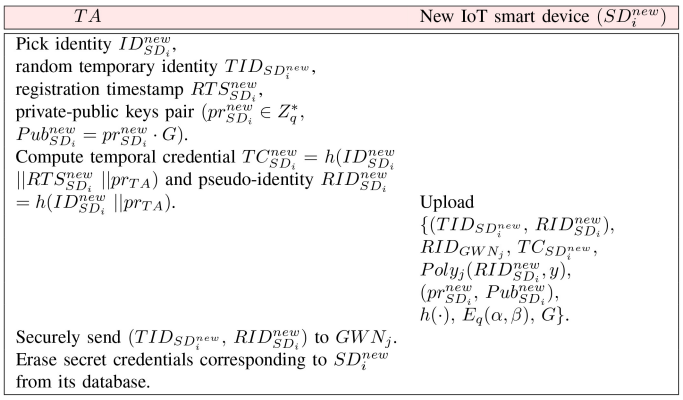


Fig. 5. Overview of dynamic nodes addition phase.

F. Dynamic Nodes Addition Phase

In this phase, to deploy a new IoT smart device, say SD_i^{new} in a group Gr_j containing the gateway node GWN_j, the *TA* requires to execute the following. At first, the *TA* selects a unique identity $ID_{SD_i}^{\text{new}}$, a unique random temporary identity $TID_{SD_i^{\text{new}}}$, a registration timestamp $RTS_{SD_i}^{\text{new}}$ and a private-public keys pair ($pr_{SD_i}^{\text{new}} \in Z_q^*$, $Pub_{SD_i}^{\text{new}} = pr_{SD_i}^{\text{new}} \cdot G$), and computes the temporal credential as $TC_{SD_i}^{\text{new}} = h(ID_{SD_i}^{\text{new}} ||RTS_{SD_i}^{\text{new}} ||pr_{TA})$ and pseudo identity $RID_{SD_i}^{\text{new}} = h(ID_{SD_i}^{\text{new}} ||pr_{TA})$. The *TA* then uploads the information $\{(TID_{SD_i^{\text{new}}}, RID_{SD_i}^{\text{new}}), RID_{\text{GWN}_j}, TC_{SD_i^{\text{new}}}, \text{Poly}_j(RID_{SD_i}^{\text{new}}, y), (pr_{SD_i}^{\text{new}}, Pub_{SD_i}^{\text{new}}), h(\cdot), E_q(\alpha, \beta), G\}$ into the memory of SD_i^{new} prior to its deployment in Gr_j. In addition, the *TA* also sends securely $(TID_{SD_i^{\text{new}}}, RID_{SD_i}^{\text{new}})$ to GWN_j and deletes the secret credentials corresponding to SD_i^{new} from its database. Finally, this phase is summarized in Fig. 5.

G. Key Revocation and Node Deletion Phase

As discussed in the threat model discussed in Section I-A2, "some IoT smart devices can be physically captured by an adversary $\mathcal{A}$, and the credentials stored in the captured devices are compromised using the power analysis attacks" [10]. We assume that a gateway node GWN_j can detect a physically captured smart device SD_i through some intrusion detection mechanisms. To revoke the credentials (keys) and delete SD_i from the network, the gateway node GWN_j removes the credentials related to SD_i from its secure database. The GWN_j prepares a revocation list RL_{SD} which is initially set to empty, that is, $RL_{SD} = \emptyset$, and includes RID_{SD_i} in it, that is, $RL_{SD} = RL_{SD} \cup \{RID_{SD_i}\}$. Moreover, a fake IoT smart device, say SD_i^f can not be deployed with the same credentials as RL_{SD} contains RID_{SD_i}. In addition, $TC_{SD_i}^f = h(ID_{SD_i}^f ||RTS_{SD_i}^f ||pr_{TA})$ and $RID_{SD_i}^f = h(ID_{SD_i}^f ||pr_{TA})$ require the private key pr_{TA} of the *TA*.

TABLE III
VARIOUS QUERIES AND THEIR SIGNIFICANT

Query	Purpose
$Execute(\lambda_{SD_i}^{c1}, \lambda_{GWN_j}^{c2})$	Under this query, $\mathcal{A}$ has ability to eavesdrop communicated messages between SD_i and GWN_j
$CorruptSD(\lambda_{SD_i}^{c})$	By executing such a query, $\mathcal{A}$ is able to extract all the per-stored secret credentials of a compromised SD_i.
$Reveal(\lambda^c)$	$\mathcal{A}$ executes this query in order to disclose the session key shared between λ^c and its respective participant
$Test(\lambda^c)$	This query permits $\mathcal{A}$ to verify the session key between SD_i and GWN_j to check if the derived session key is original one or just a random key

IV. SECURITY ANALYSIS

Wang *et al.* [52] analyzed several authentication schemes and came up with an important observation that the widely utilized formal techniques, including the random oracle model and Burrows–Abadi–Needham (BAN) logic [53] can not capture some structural mistakes in the schemes. As a result, ensuring the soundness of authentication protocols becomes an open issue. To assure the security of the proposed BSKMP-ICPS with high probability against an adversary, we evaluate for its robustness against different attacks using both the "formal (mathematical) security analysis under the broadly accepted ROR model" [14], "informal (nonmathematical) security analysis," and 'formal security verification using automated software validation tool."

A. Formal Security Under the ROR Model

In this section, we elaborate that the established session key between an IoT smart device (SD_i) and its associated gateway node (GWN_j) in the proposed scheme (BSKMP-ICPS) is formally secure against an adversary $\mathcal{A}$ under the widely accepted the "ROR oracle model" [14]. The proof of the session key security of BSKMP-ICPS mentioned in Theorem 1 has been done under the semantic security notion as defined in Definition 1. The "one-way cryptographic hash function" $h(\cdot)$ is considered as a random oracle, say *Hash*. In addition, the adversary $\mathcal{A}$ has access to the queries as tabulated in Table III.

A smart device SD_i and a gateway node GWN_j are two entities, called participants, which are involved during the key management phase. Let $\lambda_{SD_i}^{c1}$ and $\lambda_{\text{GWN}_j}^{c2}$ denote c_1 and c_2 instances of SD_i and GWN_j, respectively, which are known as "random oracles." An instance λ^{c_1} or λ^{c_2} is said to be *fresh*, if the adversary $\mathcal{A}$ cannot disclose the session key SK_{SD_i,GWN_j} $(= SK_{\text{GWN}_j,SD_i})$ assisted by the $Reveal(\lambda^c)$ query as mentioned in Table III.

Definition 1 (Semantic Security): Let $Adv_{\mathcal{A}}^{BSKMP-ICPS}(t)$ refer to the "advantage of an adversary $\mathcal{A}$, running in polynomial time t_p for breaking the semantic security of the proposed BSKMP-ICPS in order to derive the session key SK_{SD_i,GWN_j} $(= SK_{\text{GWN}_j,SD_i})$ between a smart device SD_i and a gateway node GWN_j." Then, $Adv_{\mathcal{A}}^{BSKMP-ICPS}(t_p) = |2Pr[b' = b] - 1|$, where b and b' are the "correct" and "guessed" bits, respectively.

Theorem 1: Let an adversary $\mathcal{A}$ executing in polynomial time t_p attempt to obtain the session key SK_{SD_i,GWN_j} $(= SK_{\text{GWN}_j,SD_i})$ established between a smart device SD_i and a gateway node GWN_j for a particular session in the proposed BSKMP-ICPS during the key management phase. If q_h, $|Hash|$ and $Adv_{\mathcal{A}}^{ECDDHP}(t_p)$ represent the "number of *Hash* queries," the "range space of a one-way collision-resistant hash function $h(\cdot)$," and the "advantage in breaking the Elliptic Curve Decisional Diffie–Hellman Problem (ECDDHP)," respectively, then $Adv_{\mathcal{A}}^{BSKMP-ICPS}(t_p) \leq (q_h^2/|Hash|) + 2Adv_{\mathcal{A}}^{ECDDHP}(t_p)$.

Proof: The proof of this theorem is similar to that as mentioned in [54]–[56]. A sequence of games is used to prove the security of the proposed BSKMP-ICPS. In BSKMP-ICPS, the adversary $\mathcal{A}$ executes three games, say $\text{Game}_i^{\mathcal{A}}$, $(i = 0, 1, 2)$, where $Success_{\text{Game}_i}^{\mathcal{A}}$ represents an event in which $\mathcal{A}$ can accurately guess a random bit b in $\text{Game}_i^{\mathcal{A}}$. Thus, $\mathcal{A}$'s advantage (success probability) to win $\text{Game}_i^{\mathcal{A}}$ becomes $Adv_{\mathcal{A},\text{Game}_i}^{BSKMP-ICPS} = Pr[Success_{\text{Game}_i}^{\mathcal{A}}]$. The detailed description of each game is provided below.

$\text{Game}_0^{\mathcal{A}}$: In this game, $\mathcal{A}$ plays the actual attack against BSKMP-ICPS and picks a random bit b prior to the starting of the game $\text{Game}_0^{\mathcal{A}}$. By the "semantic security definition defined in Definition 1," it follows that:

$$Adv_{\mathcal{A}}^{BSKMP-ICPS}(t_p) = \left|2Adv_{\mathcal{A},\text{Game}_0}^{BSKMP-ICPS} - 1\right|. \tag{1}$$

$\text{Game}_1^{\mathcal{A}}$: In this game, "$\mathcal{A}$ plays an eavesdropping attack by performing the *Execute* query followed by the *Test* query." The result of the *Test* query helps $\mathcal{A}$ to decide whether he/she gets an original session key SK_{SD_i,GWN_j} $(= SK_{\text{GWN}_j,SD_i})$ or a random key. For achieving this goal, $\mathcal{A}$ needs to eavesdrop the transmitted messages $Msg_1 = \{TID_{SD_i}, TID_{SD_i}^*, RN_{SD_i}, Sig_{SD_i}, TS_{SDi1}\}$, $Msg_2 = \{RN_{\text{GWN}_j}, Sig_{\text{GWN}_j}, TS_{GWNj}\}$, and $Msg_3 = \{SKV_{SD_i,\text{GWN}_j}, TS_{SDi2}\}$ during the key management phase between SD_i and GWN_j. The derivation of the session key $SK_{SD_i,\text{GWN}_j} = h(\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j}) \,||DHK_{SD_i,\text{GWN}_j} \,||Sig_{SD_i})$, where $DHK_{SD_i,\text{GWN}_j} = x_{SD_i} \cdot RN_{\text{GWN}_j}$ requires the short term credentials $\{x_{SD_i}, y_{\text{GWN}_j}\}$ as well as long-term secrets $\{RID_{SD_i}, RID_{\text{GWN}_j}\}$. In addition, the secret parameters are protected by the "collision-resistant one-way hash function $h(\cdot)$." Therefore, only hijacking of these messages will not improve the success probability for revealing the session key SK_{SD_i,GWN_j} $(= SK_{\text{GWN}_j,SD_i})$. Thus, both the games $\text{Game}_0^{\mathcal{A}}$ and $\text{Game}_1^{\mathcal{A}}$ are "indistinguishable under the eavesdropping attack." As a result, we have

$$Adv_{\mathcal{A},\text{Game}_1}^{BSKMP-ICPS} = Adv_{\mathcal{A},\text{Game}_0}^{BSKMP-ICPS}. \tag{2}$$

$\text{Game}_2^{\mathcal{A}}$: Under this game, "$\mathcal{A}$ plays an active attack, where the simulation of *Hash* queries and solution of the computational ECDDHP are executed." $\mathcal{A}$ needs to derive the session key $SK_{SD_i,\text{GWN}_j} = h(\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j}) \,||DHK_{SD_i,\text{GWN}_j} \,||Sig_{SD_i}) = h(\text{Poly}_j(RID_{\text{GWN}_j}, RID_{SD_i}) \,||DHK_{\text{GWN}_j,SD_i} \,||Sig_{SD_i}) = SK_{\text{GWN}_j,SD_i}$, where $DHK_{SD_i,\text{GWN}_j} = x_{SD_i} \cdot RN_{\text{GWN}_j}$ and $DHK_{\text{GWN}_j,SD_i} = y_{\text{GWN}_j} \cdot RN_{SD_i}$. $\mathcal{A}$ needs to know the secrets credentials $\{x_{SD_i}, y_{\text{GWN}_j}\}$ as well as the secrets $\{RID_{SD_i}, RID_{\text{GWN}_j}\}$ to derive the session key. However, these secrets are protected by the "collision-resistant one-way hash function $h(\cdot)$." Moreover, from the eavesdropped messages Msg_1, Msg_2, and Msg_3, $\mathcal{A}$ has only the knowledge of $\{RN_{SD_i}, RN_{\text{GWN}_j}\}$. To derive either DHK_{GWN_j,SD_i} or DHK_{SD_i,GWN_j}, $\mathcal{A}$ needs to solve ECDDHP

from the eavesdropped RN_{SD_i} and RN_{GWN_j}. As a result, $\mathcal{A}$ only can derive the session key if he/she can solve *Hash* query and ECDDHP. Both the games $\text{Game}_1^{\mathcal{A}}$ and $\text{Game}_2^{\mathcal{A}}$ become indistinguishable if we exclude the simulation of *Hash* queries and the computational ECDDHP. Using the "birthday paradox for finding the hash collision and the advantage of solving ECDDHP," we have the following relationship:

$$\left|Adv_{\mathcal{A},\text{Game}_1}^{BSKMP-ICPS} - Adv_{\mathcal{A},\text{Game}_2}^{BSKMP-ICPS}\right| \leq \frac{q_h^2}{2|\text{Hash}|} + Adv_{\mathcal{A}}^{ECDDHP}(t_p). \quad (3)$$

It is noted that all the queries except guessing a bit to win the game $\text{Game}_2^{\mathcal{A}}$ are executed by $\mathcal{A}$. It then follows that $Adv_{\mathcal{A},\text{Game}_2}^{BSKMP-ICPS} = (1/2)$.

Equations (1)–(3) and "the application of triangular inequality" lead to the following derivation: $(1/2).Adv_{\mathcal{A}}^{BSKMP-ICPS}(t_p) = |Adv_{\mathcal{A},\text{Game}_0}^{BSKMP-ICPS} - (1/2)| = |Adv_{\mathcal{A},\text{Game}_0}^{BSKMP-ICPS} - Adv_{\mathcal{A},\text{Game}_2}^{BSKMP-ICPS}| = |Adv_{\mathcal{A},\text{Game}_1}^{BSKMP-ICPS} - Adv_{\mathcal{A},\text{Game}_2}^{BSKMP-ICPS}| \leq (q_h^2/2|Hash|) + Adv_{\mathcal{A}}^{ECDDHP}(t_p)$. Finally, if we multiply "both sides by a factor of 2," we come to the main result: $Adv_{\mathcal{A}}^{BSKMP-ICPS}(t_p) \leq (q_h^2/|Hash|) + 2Adv_{\mathcal{A}}^{ECDDHP}(t_p)$. ■

B. Informal Security Analysis

We show the robustness of the proposed BSKMP-ICPS against the following potential attacks.

1) Impersonation Attacks: Suppose an adversary $\mathcal{A}$ has all the intercepted messages $Msg_1 = \{TID_{SD_i}, TID_{SD_i}^*, RN_{SD_i}, Sig_{SD_i}, TS_{SDi1}\}$, $Msg_2 = \{RN_{\text{GWN}_j}, Sig_{\text{GWN}_j}, TS_{GWNj}\}$, and $Msg_3 = \{SKV_{SD_i,\text{GWN}_j}, TS_{SDi2}\}$ which were exchanged during a session in the key management phase. Assume $\mathcal{A}$ wants to impersonate GWN_j on behalf of a legitimate IoT smart device SD_i by creating valid messages Msg_1 and Msg_3. For this, $\mathcal{A}$ can create its own random secrets and current timestamps, but can not create the signatures on random secret and session key because of unknown private key pr_{SD_i} of SD_i. The similar situation will occur in creating valid Msg_2 because the private key pr_{GWN_j} of GWN_j is unknown to $\mathcal{A}$. This means that both smart device and gateway impersonation attacks are resisted in BSKMP-ICPS.

2) Replay Attack: If $\mathcal{A}$ intercepts the messages Msg_1, Msg_2, and Msg_3, and tries to replay the same old messages after some time, the old messages are detected at the recipient sides because of validation of timestamps attached in the messages. Moreover, validation of signatures and session key verifier will fail because of involvement of timestamps attached in the signatures too. Thus, BSKMP-ICPS is resilient against "the relay attack."

3) Privileged-Insider Attack: In BSKMP-ICPS, the *TA* being the trusted entity is solely responsible for enrolling the IoT smart devices and the gateway nodes along with the fog and cloud servers. It is also noted that none of the registration information is sent to the *TA* by each registered SD_i and GWN_j. Rather, each SD_i and GWN_j receives securely the registration credentials from the *TA* prior to their deployment in the ICPS environment. As a result, the possibilities of mounting "privileged-insider attacks" by an adversary $\mathcal{A}$ are completed eliminated in BSKMP-ICPS.

4) Physical IoT Smart Device Capture Attack: As per the attack model discussed in Section I-A2, an adversary $\mathcal{A}$ can physically capture some IoT smart devices. Assume an IoT smart device SD_i has been physically captured by $\mathcal{A}$ and all the secret credentials stored in its memory are extracted. Since all the secret credentials are unique and distinct for all the deployed smart devices, capture of SD_i does not affect in compromising the secret keys derivation among the noncompromised smart devices and their gateway node. Moreover, for t-degree bivariate polynomial $\text{Poly}_j(x, y)$, we fix $t >> n_{sd}$. If no more than t smart devices are compromised, the original polynomial $\text{Poly}_j(x, y)$ can not be constructed using Lagrange's interpolation [57]. Thus, BSKMP-ICPS is "unconditional secure and t-collusion resistant" against smart device physical capture attack.

5) Man-in-the-Middle Attack: Suppose an adversary $\mathcal{A}$ intercepts the messages Msg_1, Msg_2, and Msg_3, and tries to modify them on the fly to convert them into valid messages so that the recipients can not detect the modified messages. To modify Msg_1, even if $\mathcal{A}$ generates a new ransom secret $rn'_{SD_i} \in Z_q^*$ and a current timestamp TS'_{SDi1} to calculate $x'_{SD_i} = h(RID_{SD_i} || rn'_{SD_i} || TC_{SD_i} || TS'_{SDi1})$ and $RN'_{SD_i} = x'_{SD_i} \cdot G$, he/she will stuck because the long-term secrets RID_{SD_i} and TC_{SD_i} are unknown, and hence, it is computationally difficult to compute RN_{SD_i}. Also, due to the intractability of ECDLP, it is difficult to compute x_{SD_i} from RN_{SD_i} which leads further difficult to derive the long-term secrets due to "collision resistant property" of $h(\cdot)$. A similar situation comes for modification of other messages Msg_2 and Msg_3. Therefore, BSKMP-ICPSs is secure against the MiTM attack.

6) Ephemeral Secret Leakage Attack: The key management phase described in Section III-C permits an IoT smart device SD_i to establish a secret key shared with its GWN_j as $SK_{SD_i,\text{GWN}_j} = h(\text{Poly}_j(RID_{SD_i}, RID_{\text{GWN}_j}) || DHK_{SD_i,\text{GWN}_j} || Sig_{SD_i})$, which is same as $SK_{\text{GWN}_j,SD_i} = h(\text{Poly}_j(RID_{\text{GWN}_j}, RID_{SD_i}) || DHK_{\text{GWN}_j,SD_i} || Sig_{SD_i})$, because $DHK_{\text{GWN}_j,SD_i} = y_{\text{GWN}_j} \cdot RN_{SD_i} = x_{SD_i} \cdot RN_{\text{GWN}_j} = DHK_{\text{GWN}_j,SD_i}$. Since the secret key SK_{GWN_j,SD_i} relies on both short term (random) secrets and long term (secret) credentials, under the CK-adversary model, an adversary can not compute the secret keys unless both types of secrets are known. Again, to derive DHK_{GWN_j,SD_i} $(= DHK_{\text{GWN}_j,SD_i})$, the adversary needs to solve the "Elliptic Curve Dicisional Diffie–Hellman Problem (ECDDHP)" from the public RN_{SD_i} and RN_{GWN_j}. Thus, ESL attack is protected in BSKMP-ICPS too.

V. Formal Security Verification Using AVISPA: Simulation Study

This section shows a simulation study on formal security verification using the widely adapted AVISPA software validation tool [15]. Any security protocol to be tested under the AVISPA tool needs to be implemented in the "high-level protocol specification language (HLPSL)," which is a role-oriented language. AVISPA contains four backends, namely: 1) "on-the-fly model-checker (OFMC)"; 2) "constraint-logic-based

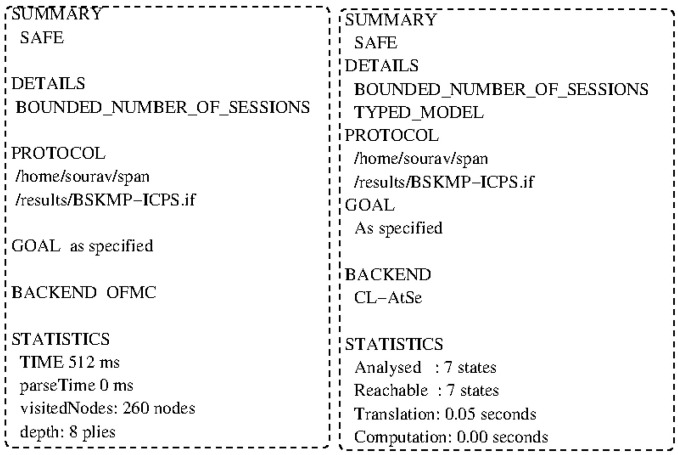


Fig. 6. Simulation results under OFMC and CL-AtSe backends.

attack searcher (CL-AtSe)"; 3) "SAT-based model checker (SATMC)"; and 4) "tree automata based on automatic approximations for the analysis of security protocols (TA4SP)."

The enrollment and key management phases have been implemented using HLPSL where the basic roles are defined for the *TA*, an IoT smart device SD_i and a gateway node GWN_j, and also two mandatory roles for the "session" and "goal and environment" are defined as the composite roles. The HLPSL code is translated to its "intermediate format (IF)" with the help of HLPSL2IF translator that is supported by AVISPA. The IF then serves as an input to one of the four available backends to produce the "output format (OF)" which precisely tells us "whether a tested security protocol is safe or unsafe." If a protocol becomes unsafe, the OF also mentions about some statistics and attack trace. More detailed description on AVISPA and its HLPSL can be found in [15].

Because AVISPA applies the DY threat model (as discussed in Section I-A2), only replay and "main-in-the-middle" attacks are detected. The proposed protocol was simulated under the "SPAN, the Security Protocol ANimator for AVISPA" [15] and the simulation results are demonstrated in Fig. 6. Under OFMC backend, the number of visited nodes was 260 with a depth of 8 plies, and the total time taken was 512 ms. On the other side, out of seven analyzed states, all the states were reachable, and the translation time was 0.05 s. The simulation results reported in Fig. 6 clearly show that the protocol is secure against both replay and "main-in-the-middle" attacks.

VI. Experimental Results Using MIRACL

We perform experiments of various cryptographic primitives using the broadly accepted MIRACL [16].

We denote the notations T_{ecm}, T_{eca}, $T_{\text{senc}}/T_{\text{sdec}}$, T_{exp}, T_h, T_{mul}, T_{add}, $T_{ibe-keygen}$, $T_{ibe-enc}$, and $T_{ibe-dec}$ as the time needed for "elliptic curve point (scalar) multiplication," "elliptic curve point addition," "symmetric key [Advanced Encryption Standard (AES-128)] encryption/decryption," "modular exponentiation," "one-way hash function using the SHA-256 hashing algorithm," "modular multiplication over $GF(q)$," "modular addition over $GF(q)$," "IBE key generation," "IBE encryption," and "IBE decryption," respectively. Moreover, we considered a nonsingular elliptic curve of the type: "$y^2 = x^3 + \alpha x + \beta \pmod q$." In addition, T_{ecenc}, T_{ecdec}, T_{ecsigg}, and T_{ecsigv} represent the time required for "ECC-based encryption," "ECC-based decryption," "ECDSA signature generation" and "ECDSA signature verification," respectively.

TABLE IV
Execution Time (in Milliseconds) Using MIRACL

Primitive	Max. time (ms)	Min. time (ms)	Average time (ms)
T_h	0.149	0.024	0.055
T_{exp}	0.248	0.046	0.072
T_{ecm}	2.998	0.284	0.674
T_{eca}	0.002	0.001	0.002
T_{ecenc}	5.998	0.569	1.350
T_{ecdec}	3.000	0.285	0.676
T_{ecsigg}	3.147	0.308	0.729
T_{ecsigv}	6.147	0.593	1.405
T_{senc}	0.003	0.001	0.001
T_{sdec}	0.002	0.001	0.001
T_{mul}	0.007	0.001	0.002
T_{add}	0.003	0.001	0.001
$T_{ibe-keygen}$	0.072	0.070	0.071
$T_{ibe-enc}$	3.214	2.627	2.753
$T_{ibe-dec}$	9.189	8.664	8.707

The following platform has been considered for MIRACL: "Ubuntu 18.04.4 LTS, with memory: 7.7 GiB, processor: Intel Core i7-8565U CPU @ 1.80 GHz × 8, OS type: 64 bit and disk: 966.1 GB." All the experiments have been run for each cryptographic primitive for 100 times. In addition, "maximum, minimum, and average run-time in milliseconds have been calculated for each cryptographic primitive from the 100 runs," and are then briefed in Table IV.

VII. Comparative Analysis

We provide a comparative analysis of the proposed BSKMP-ICPS with the existing recent relevant schemes designed by Harishma *et al.* [28] and Lara *et al.* [33].

A. Security and Functionality Features

The "security and functionality features comparison" among BSKMP-ICPS and other schemes [28], [33] is shown in Table V. We have considered 13 features ($SecFA_1$–$SecFA_{13}$) in the comparison. We have applied the existing evaluation criteria discussed in Section I-B along with our own evaluation metrics. It is noticed that Harishma *et al.*'s scheme [28] does not support the features $SecFA_8$–$SecFA_{10}$ and $SecFA_{13}$, whereas Lara *et al.*'s scheme [33] also fails to support the features $SecFA_{10}$–$SecFA_{13}$. On the other side, BSKMP-ICPS provides "better security and more functionality features ($SecFA_1$–$SecFA_{13}$)" as compared to the schemes [28], [33].

B. Computational Costs

We use the experimental results reported in Section VI under the MIRACL for estimation of computation time needed by BSKMP-ICPS and other schemes [28], [33].

We consider key management/authentication phase for all the considered schemes. In BSKMP-ICPS, during key management phase, an IoT smart device SD_i and its associated

TABLE V
COMPARATIVE STUDY ON SECURITY AND FUNCTIONALITY ATTRIBUTES

Feature	Harishma *et al.* [28]	Lara *et al.* [33]	BSKMP-ICPS
$SecFA_1$	✓	✓	✓
$SecFA_2$	✓	✓	✓
$SecFA_3$	×	✓	✓
$SecFA_4$	✓	✓	✓
$SecFA_5$	✓	✓	✓
$SecFA_6$	✓	✓	✓
$SecFA_7$	✓	✓	✓
$SecFA_8$	×	✓	✓
$SecFA_9$	×	✓	✓
$SecFA_{10}$	×	×	✓
$SecFA_{11}$	✓	×	✓
$SecFA_{12}$	✓	×	✓
$SecFA_{13}$	×	×	✓
$SecFA_{14}$	×	×	✓

Note: $SecFA_1$: "resistant to privileged insider attack"; $SecFA_2$: "replay attack"; $SecFA_3$: "man-in-the-middle attack"; $SecFA_4$: "mutual authentication"; $SecFA_5$: "key agreement"; $SecFA_6$: "device/gateway node/server impersonation attack"; $SecFA_7$: "resilience against device physical capture attack"; $SecFA_8$: "session key security under the CK-adversary model"; $SecFA_9$: "formal security verification using AVISPA tool"; $SecFA_{10}$: "dynamic node addition phase"; $SecFA_{11}$: "IoT smart device/server anonymity property"; $SecFA_{12}$: "untraceability property"; $SecFA_{13}$: "key revocation and node deletion phase"; $SecFA_{14}$: "support to blockchain-based solution". ✓: "a scheme is secure or it assists a feature"; ×: "a scheme is insecure or it does not assist a feature".

TABLE VI
COMPARATIVE COMPUTATIONAL COSTS ANALYSIS

Scheme	Computational cost	Estimated time (in ms)
BSKMP-ICPS	$12T_h + 8T_{ecm} + 2T_{eca} + 2T_{poly}$	6.656
Harishma *et al.*	$11T_h + 4T_{exp} + T_{senc} + T_{ibe-keygen} + T_{ibe-dec}$	9.672
Lara *et al.*	$16T_h$	0.880

gateway node GWN_j need the computational costs of $6T_h + 4T_{\text{ecm}} + T_{\text{eca}} + T_{\text{poly}}$ and $6T_h + 4T_{\text{ecm}} + T_{\text{eca}} + T_{\text{poly}}$, respectively, where T_{poly} denotes the time required to evaluate a t-degree polynomial over the finite field GF_q. If we apply Horner's rule [58] for polynomial evaluation, we have $T_{\text{poly}} = t(T_{\text{mul}} + T_{\text{add}})$. Considering $t = 100$, the total computation cost in BSKMP-ICPS becomes $12T_h + 8T_{\text{ecm}} + 2T_{\text{eca}} + 2T_{\text{poly}} \approx 6.656$ ms. The comparative analysis on "computational costs" among BSKMP-ICPS and other schemes [28], [33] shown in Table VI demonstrates that BSKMP-ICPS requires less computational cost as compared to that for Harishma *et al.*'s scheme [28]. Although BSKMP-ICPS requires more computational cost as compared to that for Lara *et al.*'s scheme [33] which depends on the hash function computations only, BSKMP-ICPS provides superior security and functionality features as compared to Lara *et al.*'s scheme (see Table V). Moreover, the computational time needed by the leader node and its peer cloud server nodes in the P2P network for block verification and addition described in Algorithm 1 is shown in Table VII.

C. Communication Costs

For comparative analysis on communication costs involved in the key management phase or authentication phase of BSKMP-ICPS and other schemes [28], [33], we consider the bit sizes of "identity/temporary or pseudo identity," "hash output (using the SHA-256 hashing algorithm [59])," "elliptic curve point of the form $P = (P_x, P_y)$," "random secret," and "timestamp" as 160, 256, (160+160) = 320, 160 , and 32 bits, respectively, Furthermore, in Harishma *et al.*'s scheme [28], "authentication request," "nonce," and "challenge vector" are considered as 160, 160, and 160 bits, respectively, whereas in Lara *et al.*'s scheme [33], "pseudonym" is taken as 160 bits. It is also assumed that 160-bit ECC provides that same security level as that for 1024-bit "RSA public-key cryptosystem." In addition, to make the "discrete logarithm problem (DLP)" intractable in Harishma *et al.*'s scheme [28], the prime is chosen as 1024 bits. Under these assumptions, in BSKMP-ICPS, three messages Msg_1, Msg_2, and Msg_3 require 928, 512, and 288 bits, respectively. Summing all these terms altogether, the total communication cost involved in BSKMP-ICPS becomes 1724 bits. From Table VIII, it is a clear evidence that BSKMP-ICPS needs significantly less communication cost and the number of exchanged messages as compared to those for other schemes [28], [33]. In addition, the communication costs needed by the leader node and its peer cloud server nodes in the P2P network for block verification and addition described in Algorithm 1 are also provided in Table VII.

TABLE VII
BLOCK ADDITION AND VERIFICATION COSTS ANALYSIS

Computational costs analysis		
Leader node	Other peer node	Total estimated time
$(n_{cs}-1)[T_{ecenc} + T_{ecdec}] \approx 2.026(n_{cs}-1)$ ms	$(n_{cs}-1)[T_{ecenc} + T_{ecdec} + T_{ecsigv} + \log_2(n_t)T_h] \approx (n_{cs}-1)[3.431 + 0.055\log_2(n_t)]$ ms	$(n_{cs}-1)[2T_{ecenc} + 2T_{ecdec} + T_{ecsigv} + \log_2(n_t)T_h] \approx (n_{cs}-1)[5.457 + 0.055\log_2(n_t)]$ ms
Communication costs analysis		
Leader node	Other peer node	Total cost
$[1674 + 640n_t + 672](n_{cs}-1)$ bits	$672(n_{cs}-1)$ bits	$[3018 + 640n_t](n_{cs}-1)$ bits

TABLE VIII
COMPARATIVE COMMUNICATION COSTS ANALYSIS

Protocol	No. of messages	Total cost (in bits)
BSKMP-ICPS	3	1724
Harishma *et al.*	6	$160k + 4416$
Lara *et al.*	4	2080

k: number of challenge vectors used in Harishma *et al.*'s scheme [28].

TABLE IX
COMPARATIVE STORAGE COSTS ANALYSIS

Protocol	SC_1	SC_2
BSKMP-ICPS	$2208 + (t+1)\log_2(q)$ bits	$1472 + 416n_{sd} + (t+1)\log_2(q)$ bits
Harishma *et al.*	$160k$ bits	$160k + 2208$ bits
Lara *et al.*	$160n_{kp} + 672$ bits	$160n_{kp} + 672$ bits

SC_1: storage cost at IoT smart (sensor) device; SC_2: storage cost at gateway node/server/cloud; k: number of challenge vectors used in Harishma *et al.*'s scheme [28]; n_{kp}: number of keys in the key pool in Lara *et al.*'s scheme [33]; t: degree of bivariate polynomial in the proposed BSKMP-ICPS.

D. Storage Costs

We consider the credentials stored in the IoT smart devices and servers for calculating the requirement of storage costs during their registration/enrollment phases. We have consider the bit sizes of the entities that are used for calculation of communication costs in Section VII-C. The comparative storage costs analysis among the proposed BSKMP-ICPS and other

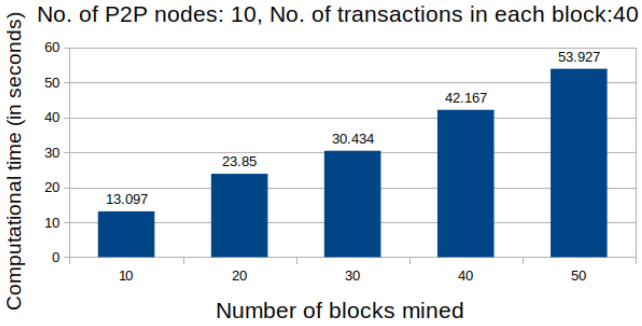


Fig. 7. Blockchain-based simulation outcomes in Scenario 1.

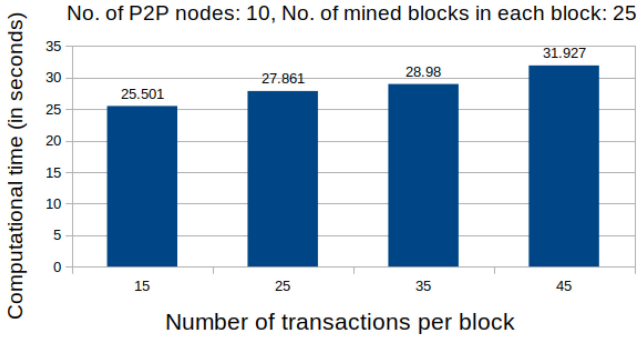


Fig. 8. Blockchain-based simulation outcomes in Scenario 2.

existing competing schemes provided in Table IX shows that the storage costs required by various entities in BSKMP-ICPS are comparable with those for other schemes.

VIII. Blockchain Implementation

This section demonstrates the blockchain implementation of the proposed scheme (BSKMP-ICPS). The cloud servers in the P2P network gather the encrypted transactions from their respective fog servers. Now, when the number of transactions comes to a predefined transactions threshold (n_t), a leader L from P2P CS network will be elected based on the round-robin policy for "blocks creation, verification and addition into the blockchain." Suppose the leader L has the block, say *FullBlk*, as demonstrated in Fig. 4, and will perform block addition into the blockchain. The "voting-based PBFT consensus algorithm provided in Algorithm 1" is then executed in order to add the created blocks by L into the blockchain.

The details simulation are provided in two scenarios as shown in Figs. 7 and 8. In both cases, we assume the total number of peer nodes in the P2P network is 10. The simulation was performed on a platform having "Ubuntu 18.04, 64-bit OS with Intel Core i5-4210U CPU @ 1.70 GHz, 4-GB RAM." The script (programming language) was written in "Node.js language with VS CODE 2019" [60]. The size of a block Fig. 4 is calculated under the following assumption: "block version, previous block hash, Merkle tree root, application type, timestamp (epoch time), owner of the block, public key of owner, an encrypted transaction, current/last block hash (using the SHA-256 hashing algorithm), and ECDSA signature are of the sizes 32, 256, 256, 32, 42, 160, 320, 640, 256, and 320 bits, respectively." Moreover, each transaction Tx_w was encrypted using ECC encryption. Note that ECC encryption leads an output consisting of two elliptic curve points. Thus, $(320 + 320) = 640$ bits are needed for an encrypted transaction. The total block size for a block *FullBlk* is then $1674 + 640n_t$ bits.

1) *Scenario 1:* In this case, we assume the number of transactions per block is 40. The simulation results provided in Fig. 7 show that "when number of blocks mined is increased, the total computational time increases linearly."
2) *Scenario 2:* In this scenario, the number of mined blocks in each chain is 25. The simulation results demonstrated in Fig. 8 show that "the total computational time increases linearly if the number of transactions per block increases during the consensus."

IX. Conclusion

This article handles an important security service in an ICPS environment by proposing a novel blockchain-enabled key management protocol (BSKMP-ICPS). The authentic and genuine data containing in the transactions of the blocks are mined and stored in the private blockchain by the fog and cloud servers. Later, the genuine data in the blocks are used to make correct predictions using AI/ML algorithms for Big data analytic purpose in order to avoid data poisoning attacks. BSKMP-ICPS is not only secure against various potential attacks, but also achieves significantly better "security and functionality features" and "less communication overhead" and "comparable computation overhead" as compared to other relevant schemes. In addition, the blockchain-based simulation on the proposed BSKMP-ICPS has been done "for measuring the computational time needed for varied number of blocks mined and also varied number of transactions per block."

Acknowledgment

The authors would like to thank the reviewers and the Associate Editor for their valuable suggestions that helped in improving the quality, readability, and presentation of this article.

References

[1] P. Leitao, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Comput. Ind.*, vol. 81, pp. 11–25, Sep. 2016.
[2] J. Lee, M. Azamfar, and J. Singh, "A blockchain enabled cyber-physical system architecture for industry 4.0 manufacturing systems," *Manuf. Lett.*, vol. 20, pp. 34–39, Apr. 2019.
[3] (2020). *Ethereum Gas*. Accessed: Nov. 2020. [Online]. Available: https://ethereum.org/en/developers/docs/gas/
[4] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.
[5] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[6] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[7] C. Cerrudo, *Why the Shellshock Bug Is Worse than Heartbleed*, MIT Technol., Cambridge, MA, USA, 2014. Accessed: Jul. 2020. [Online]. Available: https://www.technologyreview.com/2014/09/30/171109/why-the-shellshock-bug-is-worse-than-heartbleed/

[8] S. Khandelwal. (2017). *Serious Bug Exposes Sensitive Data From Millions Sites Sitting Behind CloudFlare*. Accessed: Jul. 2020. [Online]. Available: https://thehackernews.com/2017/02/cloudflare-vulnerability.html

[9] E. Bertino, N. Shang, and S. S. Wagstaff, Jr., "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE Trans. Depend. Secure Comput.*, vol. 5, no. 2, pp. 65–70, Apr.–Jun. 2008.

[10] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[11] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Vienna, Austria, 2016, pp. 1242–1254.

[12] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.

[13] H. Li, F. Li, C. Song, and Y. Yan, "Towards smart card based mutual authentication schemes in cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 7, pp. 2719–2735, 2015.

[14] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptography (PKC)*, Les Diablerets, Switzerland, 2005, pp. 65–84.

[15] AVISPA. (2019). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Oct. 2019. [Online]. Available: http://www.avispa-project.org/

[16] (2020). *MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library*. Accessed: Apr. 2020. [Online]. Available: https://github.com/miracl/MIRACL

[17] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan.–Mar. 2018.

[18] G. Mali and S. Misra, "TRAST: Trust-based distributed topology management for wireless multimedia sensor networks," *IEEE Trans. Comput.*, vol. 65, no. 6, pp. 1978–1991, Jun. 2016.

[19] R. Saha, S. Misra, and P. K. Deb, "FogFL: Fog-assisted federated learning for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8456–8463, May 2021.

[20] I. Maity, S. Misra, and C. Mandal, "CORE: Prediction-based control plane load reduction in software-defined IoT networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1835–1844, Mar. 2021.

[21] M. Wazid, A. K. Das, M. K. Khan, A. A.-D. Al-Ghaiheb, N. Kumar, and A. V. Vasilakos, "Secure authentication scheme for medicine anti-counterfeiting system in IoT environment," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1634–1646, Oct. 2017.

[22] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.

[23] L. Vegh and L. Miclea, "Enhancing security in cyber-physical systems through cryptographic and steganographic techniques," in *Proc. IEEE Int. Conf. Autom. Qual. Test. Robot.*, Cluj-Napoca, Romania, 2014, pp. 1–6.

[24] L. Vegh and L. Miclea, "Securing communication in cyber-physical systems using steganography and cryptography," in *Proc. 10th Int. Conf. Commun. (COMM)*, Bucharest, Romania, 2014, pp. 1–4.

[25] K. K. R. Choo, M. M. Kermani, R. Azarderakhsh, and M. Govindarasu, "Emerging embedded and cyber physical system security challenges and innovations," *IEEE Trans. Depend. Secure Comput.*, vol. 14, no. 3, pp. 235–236, Mar.–Jun. 2017.

[26] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel remote user authentication and key agreement scheme for mobile client-server environment," *Appl. Math. Inf. Sci.*, vol. 7, no. 4, pp. 1365–1374, 2013.

[27] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Provably secure biometric-based user authentication and key agreement scheme in cloud computing," *Security Commun. Netw.*, vol. 9, no. 17, pp. 4103–4119, 2016.

[28] B. Harishma, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay, "POSTER: Authenticated key-exchange protocol for heterogeneous CPS," in *Proc. Asia Conf. Comput. Commun. Security (ASIACCS)*, 2018, pp. 849–851.

[29] L. Vegh, "Cyber-physical systems security through multi-factor authentication and data analytics," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Lyon, France, 2018, pp. 1369–1374.

[30] 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.

[31] 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.

[32] S. Challa, A. K. Das, P. Gope, N. Kumar, F. Wu, and A. V. Vasilakos, "Design and analysis of authenticated key agreement scheme in cloud-assisted cyber-physical systems," *Future Gener. Comput. Syst.*, vol. 108, pp. 1267–1286, Jul. 2020.

[33] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. Garcia, "Lightweight authentication protocol for M2M communications of resource-constrained devices in Industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 501, 2020.

[34] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9390–9401, Sep. 2020.

[35] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing blockchain-based access control protocol in IoT-enabled smart-grid system," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5744–5761, Apr. 2021, doi: 10.1109/JIOT.2020.3030308.

[36] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar, and Y. H. Park, "Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15824–15838, Jul. 2021, doi: 10.1109/JSEN.2020.3009382.

[37] S. Saha, D. Chattaraj, B. Bera, and A. K. Das, "Consortium blockchain-enabled access control mechanism in edge computing based generic Internet of Things environment," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 6, 2020, Art. no. e3995, doi: 10.1002/ett.3995.

[38] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for Internet of Vehicles," *J. Syst. Architect.*, vol. 113, Feb. 2021, Art. no. 101877.

[39] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019.

[40] 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.

[41] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K. R. Choo, "DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2440–2452, Jan. 2020.

[42] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.

[43] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: Performance of resource-constrained IoT networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 174–183, Jan. 2021.

[44] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 7081–7093, Nov. 2020.

[45] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Security*, vol. 1, no. 1, pp. 36–63, 2001.

[46] C. Blundo, A. D. Santis], A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Inf. Comput.*, vol. 146, no. 1, pp. 1–23, 1998.

[47] L. Howells and Y. Kalfoglou. (2020). *Security Think Tank: AI Cyber Attacks Will Be a Step-Change for Criminals*. Accessed: Jul. 2020. [Online]. Available: https://www.computerweekly.com/opinion/Security-Think-Tank-AI-cyber-attacks-will-be-a-step-change-for-criminals

[48] R. Shroff. *When Blockchain Meets Artificial Intelligence*. Accessed: Jul. 2020. [Online]. Available: https://medium.com/swlh/when-blockchain-meets-artificial-intelligence-e448968d0482

[49] X. Gong, Q. Wang, Y. Chen, W. Yang, and X. Jiang, "Model extraction attacks and defenses on cloud-based machine learning models," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 83–89, Dec. 2020.

[50] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "PRADA: Protecting against DNN model stealing attacks," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, Stockholm, Sweden, 2019, pp. 512–527.
[51] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017. [Online]. Available: arxiv.abs/1712.05526.
[52] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Secure Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.
[53] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
[54] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.
[55] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment," *Comput. Commun.*, vol. 153, pp. 229–249, Mar. 2020.
[56] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled Internet of Drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.
[57] F. B. Hildebrand, *Introduction to Numerical Analysis*, 2nd ed. New York, NY, USA: Dover, 1974.
[58] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2, 3rd ed. Boston, MA, USA: Addison-Wesley, 1997.
[59] W. E. May, *Secure Hash Standard*, document FIPS PUB 180-1, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Apr. 1995. Accessed: Apr. 2020. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
[60] K. Khullar. (2019). *Implementing PBFT in Blockchain)*. Accessed: Aug. 2020. [Online]. Available: https://medium.com/coinmonks/implementing-pbft-in-blockchain-12368c6c9548

Ashok Kumar Das (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from Indian Institute of Technology (IIT) Kharagpur, Kharagpur, India, in 1998, 2000, and 2008, respectively.

He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India. His current research interests include cryptography and network security, including security in smart grid, Internet of Things, Internet of Drones, Internet of Vehicles, cyber–physical systems and cloud computing, blockchain and AI/ML security. He has authored over 275 papers in international journals and conferences in the above areas, including over 230 reputed journal papers.

Dr. Das was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of IEEE SYSTEMS JOURNAL, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *Journal of Cloud Computing* (Springer), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience), and has served as a program committee member in many international conferences. He also served as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020.

Basudeb Bera received the M.Sc. degree in mathematics and computing from Indian Institute of Technology (Indian School of Mines) Dhanbad, Dhanbad, India, in 2014, and the M.Tech. degree in computer science and data processing from Indian Institute of Technology Kharagpur, Kharagpur, India, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India.

He has published 15 papers in international journals and conferences in his research areas. His research interests are cryptography, network security and blockchain technology.

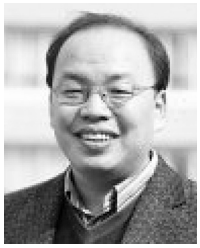
Sourav Saha (Student Member, IEEE) received the Bachelor of Technology degree in computer science and engineering from the Central Institute of Technology, Kokrajhar, India, in 2016, and the Master of Science by Research degree in computer science and engineering from Indian Institute of Information Technology Sri City, Sri City, India, in 2019. He is currently pursuing the Ph.D. degree in computer science and engineering with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India.

He has published ten papers in international journals and conferences in his research areas. His research interests include network security and blockchain technology.

Neeraj Kumar (Senior Member, IEEE) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, India, in 2009.

He was a Postdoctoral Research Fellow of Coventry University, Coventry, U.K. He is working as an Associate Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, India. He has published more than 600 technical research papers in leading journals and conferences from IEEE, Elsevier, Springer, and John Wiley.

Dr. Kumar is on the editorial board of *ACM Computing Surveys*, IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, *IEEE Network Magazine*, *IEEE Communication Magazine*, *Journal of Networks and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *International Journal of Communication Systems* (Wiley), and *Security and Privacy* (Wiley).



Ilsun You (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Fukuoka, Japan, in 2012.

He is currently a Full Professor with the Department of Information Security Engineering, Soonchunhyang University, Asan, South Korea. He has focused on 5G/6G security, security for wireless networks and mobile Internet, IoT security and so forth while publishing more than 180 papers in these areas.

Dr. You is the Editor-in-Chief of the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is on the Editorial Board of *Information Sciences*, the *Journal of Network and Computer Applications*, IEEE ACCESS, *Intelligent Automation & Soft Computing*, the *International Journal of Ad Hoc and Ubiquitous Computing*, *Computing and Informatics*, and the *Journal of High Speed Networks*. He is a Fellow of IET.

Han-Chieh Chao (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1993, respectively.

He has been the President of National Dong Hwa University, Hualien City, Taiwan, since February 2016. He has published nearly 500 peer-reviewed professional research papers. His research interests include high-speed networks, wireless networks, IPv6-based networks, and artificial intelligence.

Dr. Chao is the Editor-in-Chief of *IET Networks* and the *Journal of Internet Technology*. He has served as a Guest Editor for *ACM/Springer Mobile Networks and Applications*, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Communications Magazine*, IEEE SYSTEMS JOURNAL, *Computer Communications*, *IEEE Proceedings Communications*, *Wireless Personal Communications*, and *Wireless Communications and Mobile Computing*. He is a Fellow of IET.