Privacy-Preserving Blockchain-Based Authentication in Smart Energy Systems

Anusha Vangala*
Center for Security, Theory and Algorithmic Research,
International Institute of Information Technology
Hyderabad 500 032, India
anusha.vangala@research.iiit.ac.in

Ashok Kumar Das†
Center for Security, Theory and Algorithmic Research,
International Institute of Information Technology
Hyderabad 500 032, India
iitkgp.akdas@gmail.com,ashok.das@iiit.ac.in

ABSTRACT

Smart Energy Systems (SES) are the need of the hour, given the looming dangers of power crises amid changing climatic conditions. However, sensitive data play a critical role in such systems deserving high privacy and security protection. This paper proposes a novel blockchain-based authentication scheme that preserves privacy using the zero-knowledge protocol. During informal analysis, the proposed scheme shows resistance to various attacks such as man-in-the-middle attacks, replay attacks, impersonation attacks, privileged insider attacks, and ephemeral secret leakage attacks. The formal security verification using AVISPA regards the scheme as safe. In addition, the scheme supports critical features such as anonymity and untraceability within limited computational and communicational costs. A simulation of blockchain using Node.js shows only a linear increase in computation time with an increase in the number of blocks, and transactions, and an exponential increase with the number of nodes.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Smart Energy Systems, Authentication, Internet of Things (IoT), Blockchain, Security.

ACM Reference Format:
Anusha Vangala and Ashok Kumar Das. 2022. Privacy-Preserving Blockchain-Based Authentication in Smart Energy Systems. In *Fourth ACM International Workshop on Blockchain-enabled Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3560905.3568179

1 INTRODUCTION

A smart energy system (SES) uses various technologies of the Internet of Things (IoT), blockchain, and storage technologies to produce, conserve and efficiently manage and distribute energy from various renewable sources such as gas, hydro, thermal, and smart electricity. Recent power crises in Europe and China [1–3] show the importance of SES in managing sustainable energy. The efficient working of SES depends on the management of generated and stored data regarding user consumption, user billing, load prediction, energy demand, and supply. These data are sensitive to the users, and it is imperative to preserve their privacy while the data is used for critical decisions. In addition, these data are also vulnerable to several security issues, such as unauthorized leakage and tampering. Hence, privacy-preserving authentication protocols that use blockchain technology for tamper-free storage and management of sensitive energy data should be designed.

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568179

2 RELATED WORK

This section studies recent authentication and key agreement (AKA) schemes in the domain of smart grid and energy systems. Qi and Chen *et al.* [24] propose an ECC-based authentication scheme that preserves privacy. However, it lacks anonymity, untraceability, and support for blockchain and succumbs to ephemeral secret leakage (ESL) attacks, physical device capture, and privileged insider attacks. Sadhukhan *et al.* [34] is another scheme using ECC and symmetric cryptography that does not support blockchain technology or privacy preservation in addition to vulnerability to physical device capture and privileged insider attack and ephemeral secret leakage attack. Xiang *et al.* [34] is a bilinear pairing-based scheme that uses the zero-knowledge protocol and homomorphic encryption for privacy preservation with high computation cost and no blockchain support. The hashing-based scheme by Agilandeeswari *et al.* [7] is a simple scheme that uses secure and public channels during authentication. The security of the session key depends only on a pre-loaded key and nonces. It does not support anonymity, untraceability, or blockchain and is vulnerable to ESL, physical device capture, privileged insider, and impersonation attacks. Park *et al.* [20] propose a privacy-preserving scheme with blockchain for demand-response management. Zhu *et al.* [35] propose a privacy-preserving model to achieve authenticated data aggregation. Singh *et al.* [27] propose a blockchain-based data aggregation model that provides privacy preservation using homomorphic encryption.

3 SYSTEM MODELS

3.1 Network Models

The proposed network model consists of an energy supplier (*ES*) taking on the role of a trusted third party. The energy supplier

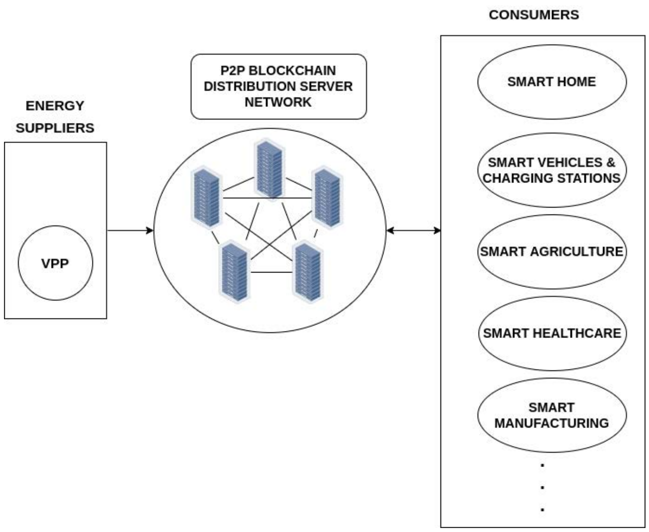

Figure 1: Blockchain-based Smart Energy Systems

passes information regarding available energy to a distribution server (*DS*) in the P2P blockchain network. The *DS* is associated with a set of consumer sensor nodes (*CN*) in the various smart consumer networks during registration. The DS knows the energy requirements of its various consumers and their energy constraints, based on which it automatically disseminates energy promptly. The energy requirements and constraints are associated with sensitive consumer information that needs privacy preservation. The private blockchain ledger is used to store details of energy management from *CN* in the various smart consumer networks. The DS collects the sensor data securely using the session key established in the proposed scheme, creates transactions and blocks, and adds them to the blockchain. The blockchain is maintained by a peer-to-peer network of trusted cloud servers that act as distribution servers.

3.2 Threat Models

A smart energy system (SES) is vulnerable to adversaries as defined in the Dolev-Yao (DY) and Honest-but-Curious (HBC) threat models [13]. The DY model gives adversary A strong capabilities as follows: 1) A has complete control over the communication channels, 2) A can eavesdrop, modify, block, replay or fabricate communication messages, 3) A can impersonate the DS or CN in consumer networks, 4) A participates in simultaneous multiple executions of the protocol, and 5) A is stateful.

The HBC model [22] gives adversaries capabilities as follows: 1) A is one of the participating entities that execute the protocol honestly without any deviation, 2) A intercepts only the messages that it sends and receives, unlike a passive DY adversary that intercepts all messages in the system, 3) A applies deductions on the stored messages to obtain information about the network and its entities, and 4) A can link the messages to deduce any information about the network and its entities.

A smart energy system records measurements frequently from the consumers and sends them to the distribution server. The information from the distribution servers is used for moderating energy supply, user-based customer billing, and load prediction. Such fine-grained data about energy usage reveals sensitive data about user behavior, raising privacy issues. Privacy breaches may lead to data leaks and user information leaks. Security breaches lead to financial loss and cyber attacks against the system. This calls for addressing several security issues, such as mutual authentication, anonymity, untraceability, and forward security.

The proposed network model has an external DY adversary, and the consumer sensor nodes can be a possible internal DY adversary. The ES is fully trusted and does not act like any adversary. The DS may be an HBC adversary but not a DY adversary as it will collapse the system [23].

4 PROPOSED SCHEME

An authentication scheme designed for smart energy systems can achieve privacy preservation in the following ways:

1) *Encryption*: This is a traditional technique to preserve data privacy. Homomorphic encryption and lattice-based cryptography are standard techniques applied to achieve privacy preservation.

2) *Anonymization*: This process applies masking, swapping, generalization, and perturbation to remove any association between collected data and the entity that owns/uses it. However, there are several de-anonymization techniques to re-identify the association. In addition, deletion of the association can reduce the usability and applicability of the data as the original data is modified [19, 21].

3) *Differential privacy*: This process extracts patterns and salient information from the collected data. Anonymization, differential identifiability, and member privacy are used over databases. Techniques such as distribution optimization and sensitivity calibration are used for differential privacy optimization, and synopsis and correlation exploitation are used in datasets. Laplacian, Gaussian, and exponential functions are used to add noise in data perturbation mechanisms for differential privacy. [16]

4) *Information theoretic privacy*: It uses entropy-based techniques such as Shannon entropy to limit the probability of disclosure of sensitive information after quantifying the data's confidentiality level [16].

5) *Zero knowledge protocols (ZKP)*: This cryptographic technique allows a party to prove the possession or knowledge of a data value without revealing it. This intelligent technique, first proposed by Goldwasser, Micali and Rackoff in 1989 [14, 15, 26]

In this paper, we use ZKP to preserve the privacy of the consumer networks' sensor nodes and the distribution server. The proposed scheme uses a signature generated by Elliptic Curve Digital Signature Algorithm (ECDSA) [17] over which ZKP is applied for privacy preservation. The proposed scheme SESAKA first registers the *CN* and *DS*, followed by the authentication and key agreement phase.

4.1 System Initialization Phase

- Step SIP_1: The ES selects a non-singular elliptic curve $E_q(a, b)$: $y^2 = x^3 + ax + b \pmod q$ over the Galois field $GF(q)$, with a "point at infinity (zero point)" O, constants $a, b \in Z_q = \{0, 1, 2, \cdots, q-1\}$ such that $4a^3 + 27b^2 \neq 0 \pmod q$ is satisfied. The ES picks a base point $G \in E_q(a, b)$ whose order n_G is as large as q, that is, $n_G \cdot G = G + G + \cdots + G$ $(n_G times) = O$, the point at infinity or zero point

- Step SIP_2: The ES picks a "collision-resistant one-way cryptographic hash function", say $H(\cdot)$ (for instance, SHA-1 hash algorithm may be used). It also selects the algorithm Practical Byzantine Fault Tolerance ($PBFT$) to be used in the consensus process in the P2P blockchain distribution server network.
- Step SIP_3: The ES picks its own master key mk_{ES} in Z_q^* and publishes the master key mk_{ES} and domain parameters $\{E_q(a,b)$, G, $H(\cdot)\}$ as public.

4.2 Registration of Consumer Smart Nodes Phase

ES registers the consumer smart nodes (CN) in the various consumer networks using this phase in offline mode.
Step CNR_1: The ES picks a true identity ID_C and temporary identity TID_C, and a random secret $c_1 \in Z_q^*$ to compute a pseudo-identity for CN as $RID_C = H(ID_C || c_1 || mk_{ES})$.
Step CNR_2: The ES picks a random private key $pr_C \in Z_q^*$, and computes the corresponding public key $Pub_C = pr_C \cdot G$ and a temporal credential for the CN as $TC_C = H(RID_C || pr_C || mk_{ES} || RTS_C)$ where RTS_C is the current timestamp of registration of CN.
Step CNR_3: The ES preloads CN with the credentials $\{(RID_C, TID_C, TC_C), H(\cdot), E_q(a,b), G, (pr_C, Pub_C)\}$. In addition, the ES publishes Pub_C as CN's public key.

4.3 Registration of Distribution Server Phase

ES uses this phase to register the distribution servers offline.
Step SR_1: The ES picks the true identity for a distribution server DS as ID_D, its temporary identity as TID_D. The ES picks random secret $d_1 \in Z_q^*$ and computes the pseudo-identity as $RID_D = H(ID_D || d_1 || mk_{ES} || RTS_D)$ where RTS_D is the registration timestamp of DS respectively.
Step SR_2: The ES preloads DS with the credentials $\{(RID_D, TID_D)$, $\{(RID_C, TID_C, TC_C)\}$, $H(\cdot)$, $E_q(a,b)$, $G\}$. After that DS picks its own random private key $pr_D \in Z_q^*$ and computes the corresponding public key $Pub_D = pr_D \cdot G$. The DS adds its private and public key (pr_G, Pub_G) to its tamper-proof secure memory database as $\{(RID_D, TID_D), \{(RID_C, TID_C, TC_C)\}, (pr_D, Pub_D), H(\cdot), E_q(a,b), G\}$. Furthermore, the DS publishes Pub_D as its public key.

4.4 Authentication and Key Agreement Phase

This phase of the SESAKA scheme is executed between the CN and DS to authenticate each other using Elliptic Curve Cryptographic (ECC) operations, hash functions, ECDSA signature, and ZKP.
Step $SESAKA_1$: CN chooses private random secrets $e_C, a_C \in Z_q^*$ and a timestamp TS_C. It computes a point $E_C = H(TID_C || e_C || TS_C) \cdot G$. This point is used for session key computation, and its privacy is preserved using ZKP such that given E_C, e_c is known without revealing it. CN also computes $t_C = H(pr_C || e_C || RID_C || TS_C) \oplus H(TC_C || TID_C || RID_C || TS_C)$, where the first hash is part of the session key and is hidden using XOR with the second hash.
Step $SESAKA_2$: CN computes the point $A_C = a_C \cdot G = (x_C, y_C)$ and assigns the first part of signature $S_{1C} = x_C$. The second part of the signature is computed as $S_{2C} = a_C^{-1}[H(pr_C || e_C || RID_C || TS_C) + H(TID_C || e_C || TS_C) \cdot S_{1C}]$. CN also computes $Q_C = S_{2C} \cdot A_C$, chooses a'_C and computes $A'_C = a'_C \cdot A_C = (x'_C, y'_C)$. For privacy preservation of the signature using ZKP, S_{2C} is sent hidden as $S'_{2C} = a_C'^{-1}[H(pr_C || e_C || RID_C || TS_C) + x'_C \cdot S_{2C}]$. CN sends $MSG_{CD1} = \langle TID_C, t_C, A_C, Q_C, A'_C, S'_{2C}, S_{1C}, E_C, TS_C \rangle$ to DS.
Step $SESAKA_3$: DS receives Msg_{CD1} at TS'_C and verifies the timestamp as $|TS'_C - TS_C| \leq \Delta T$. If true, it extracts RID_C based on TID_C from its database and extracts $H(pr_C || e_C || RID_C || TS_C) = t_C \oplus H(TC_C || TID_C || RID_C || TS_C)$. The signature is verified in two condition checks as $Q_C \stackrel{?}{=} H(pr_C || e_C || RID_C || TS_C) \cdot G + S_{1C} \cdot E_C$ and $S'_{2C} \cdot A'_C \stackrel{?}{=} H(pr_C || e_C || RID_C || TS_C) \cdot A_C + x'_C \cdot Q_C$. If both the conditions satisfy, DS chooses private random secrets $f_D, b_D \in Z_p^*$ and a timestamp TS_D. It computes a point $F_D = H(TID_D || f_D || TS_D) \cdot G$. This point is used for session key computation, and its privacy is preserved using ZKP such that given F_D, f_D is known without revealing it. DS also computes $v_D = H(pr_D || f_D || RID_D || TS_D) \oplus H(TID_C || TID_D || TC_C || TS_C || TS_D)$, where the first hash is part of the session key and is hidden using XOR with the second hash. DS computes Diffie-Hellman parameter as $DK_{DC} = H(TID_D || f_D || TS_D) \cdot E_C$ and the session key is computed as $SK_{DC} = H(DK_{DC} || H(pr_C || e_C || RID_C || TS_C) || H(pr_D || f_D || RID_D || TS_D))$.
Step $SESAKA_4$: DS generates a new temporary identity for CN as TID_C^{new} and hides it is $TID_C^* = TID_C^{new} \oplus H(SK_{DC} || TS_D)$. DS computes the point $B_D = b_D \cdot G = (x_D, y_D)$ and assigns the first part of signature $S_{1D} = x_D$. The second part of the signature is computed as $S_{2D} = b_D^{-1}[H(pr_D || f_D || RID_D || TS_D) + H(TID_D || f_D || TS_D) \cdot S_{1D}]$. DS also computes $Q_D = S_{2D} \cdot B_D$, chooses b'_D and computes $B'_D = b'_D \cdot B_D = (x'_D, y'_D)$. For privacy preservation of the signature using ZKP, S_{2D} is sent hidden as $S'_{2D} = b_D'^{-1}[H(pr_D || f_D || RID_D || TS_D) + x'_D \cdot S_{2D}]$. DS sends $MSG_{CD2} = \langle RID_D, TID_C^*, v_D, B_D, Q_D, B'_D, S'_{2D}, S_{1D}, F_D, TS_D \rangle$ to CN.
Step $SESAKA_5$: CN receives Msg_{CD2} at TS'_D and verifies the timestamp as $|TS'_D - TS_D| \leq \Delta T$. If true, it extracts $H(pr_D || f_D || RID_D || TS_D) = v_D \oplus H(TID_C || TID_D || TC_C || TS_C || TS_D)$. The signature is verified in two condition checks as $Q_D \stackrel{?}{=} H(pr_D || f_D || RID_D || TS_D) \cdot G + S_{1D} \cdot F_D$ and $S'_{2D} \cdot B'_D \stackrel{?}{=} H(pr_D || f_D || RID_D || TS_D) \cdot B_d + x'_D \cdot Q_D$. If both the conditions satisfy, CN computes Diffie-Hellman parameter as $DK_{CD} = H(TID_C || e_C || TS_C) \cdot F_D$ and the session key is computed as $SK_{CD} = H(DK_{CD} || H(pr_C || e_C || RID_C || TS_C) || H(pr_D || f_D || RID_D || TS_D))$.
Step $SESAKA_6$: CN generates a timestamp TS_{CD} and computes the session key verifier as $SKV_{CD} = H(SK_{CD} || TS_{CD})$ and sends $MSG_{CD3} = \langle SKV_{CD}, TS_{CD} \rangle$.
Step $SESAKA_7$: DS receives MSG_{CD3} at TS'_{CD} and verifies the timestamp as $|TS'_{CD} - TS'_{CD}| \leq \Delta T$. If trues, it verifies if $SKV_{CD} \stackrel{?}{=} H(SK_{DC} || TS_{CD})$. If true, it generates a new temporary identity for itself as $TID_D^{new} = H(SK_{DC} || TS_{CD})$ and updates TID_D with TID_D^{new}. DS stores the session key SK_{DC}. CN extracts its new temporary identity as $TID_C^{new} = TID_C^* \oplus H(SK_{CD} || TS_D)$ and updates TID_C with TID_C^{new} in its database. CN stores the session key SK_{CD}.
The summary of this scheme is shown in Fig 2.

4.5 Secure Data Aggregation and Blockchain Management Phase

The DS in the P2P Blockchain Distribution Server Network uses this phase after the authentication, and key agreement phase in

SESAKA Scheme	
Consumer Smart Node (CN)	Distribution Server (DS)
CN chooses e_C, $a_C \in Z_p^*$,timestamp TS_C computes $E_C = H(TID_C \Vert e_C \Vert TS_C) \cdot G$ $t_C = H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) \oplus$ $H(TC_C \Vert TID_C \Vert RID_C \Vert TS_C)$ $A_C = a_C \cdot G = (x_C, y_C)$ and assigns $S_{1C} = x_C$ $S_{2C} = a_C^{-1}[H(pr_C \Vert e_C \Vert RID_C \Vert TS_C)$ $+ H(TID_C \Vert e_C \Vert TS_C) \cdot S_{1C}]$ $Q_C = S_{2C} \cdot A_C$ chooses a'_C and computes $A'_C = a'_C \cdot A_C = (x'_C, y'_C)$ $S'_{2C} = a_C'^{-1}[H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) + x'_C \cdot S_{2C}]$ $MSG_{CD1} = \langle TID_C, t_C, A_C, Q_C, A'_C, S'_{2C}, S_{1C}, E_C, TS_C \rangle$ $\longrightarrow$ via public channel	
	$\lvert TS'_C - TS_C \rvert \leq \Delta T$. If true, Extracts RID_C using TID_C $H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) = t_C \oplus H(TC_C \Vert TID_C \Vert RID_C \Vert TS_C)$ $Q_C \stackrel{?}{=} H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) \cdot G + S_{1C} \cdot E_C$ $S'_{2C} \cdot A'_C \stackrel{?}{=} H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) \cdot A_C + x'_C \cdot Q_C$ chooses $f_D, b_D \in Z_p^*$ and a timestamp TS_D $F_D = H(TID_D \Vert f_D \Vert TS_D) \cdot G$ $v_D = H(pr_D \Vert f_D \Vert RID_D \Vert TS_D) \oplus H(TID_C \Vert TID_D \Vert$ $TC_C \Vert TS_C \Vert TS_D)$ $DK_{DC} = H(TID_D \Vert f_D \Vert TS_D) \cdot E_C$ $SK_{DC} = H(DK_{DC} \Vert H(pr_C \Vert e_C \Vert RID_C \Vert TS_C) \Vert H(pr_D \Vert$ $f_D \Vert RID_D \Vert TS_D))$ generates TID_C^{new}, computes $TID_C^* = TID_C^{new} \oplus H(SK_{DC} \Vert TS_D)$. $B_D = b_D \cdot G = (x_D, y_D)$ and assigns $S_{1D} = x_D$. $S_{2D} = b_D^{-1}[H(pr_D \Vert f_D \Vert RID_D \Vert TS_D) +$ $H(TID_D \Vert f_D \Vert TS_D) \cdot S_{1D}]$ $Q_D = S_{2D} \cdot B_D$, chooses b'_D and computes $B'_D = b'_D \cdot B_D = (x'_D, y'_D)$ $S'_{2D} = b_D'^{-1}[H(pr_D \Vert f_D \Vert RID_D \Vert$ $TS_D) + x'_D \cdot S_{2D}]$. $Msg_{CD_2} = \langle RID_D, TID_C^*, v_D, B_D, Q_D, B'_D, S'_{2D}, S_{1D}, F_D, TS_D \rangle$ $\longleftarrow$ via public channel
$\lvert TS'_D - TS_D \rvert \leq \Delta T$. If true, $H(pr_D \Vert f_D \Vert RID_D \Vert TS_D) = v_D \oplus$ $H(TID_C \Vert TID_D \Vert TC_C \Vert TS_C \Vert TS_D)$ $Q_D \stackrel{?}{=} H(pr_D \Vert f_D \Vert RID_D \Vert TS_D) \cdot G + S_{1D} \cdot F_D$ $S'_{2D} \cdot B'_D \stackrel{?}{=} H(pr_D \Vert f_D \Vert RID_D \Vert TS_D) \cdot B_d + x'_D \cdot Q_D$ $DK_{CD} = H(TID_C \Vert e_C \Vert TS_C) \cdot F_D$ $SK_{CD} = H(DK_{CD} \Vert H(pr_C \Vert e_C \Vert RID_C \Vert$ $TS_C) \Vert H(pr_D \Vert f_D \Vert RID_D \Vert TS_D))$ generates timestamp TS_{CD}, computes $SKV_{CD} = H(SK_{CD} \Vert TS_{CD})$ $Msg_{CD_3} = \langle SKV_{CD}, TS_{CD} \rangle$ $\longrightarrow$ via open channel	
	$\lvert TS'_{CD} - TS'_{CD} \rvert \leq \Delta T$. If true, $SKV_{CD} \stackrel{?}{=} H(SK_{DC} \Vert TS_{CD})$. If true, $TID_D^{new} = H(SK_{DC} \Vert TS_{CD})$ updates TID_D with TID_D^{new} stores SK_{DC}
$TID_C^{new} = TID_C^* \oplus H(SK_{CD} \Vert TS_D)$ updates TID_C with TID_C^{new} stores SK_{CD}	
Both CN and DS share the same secret key $SK_{CD} = SK_{DC}$	

Figure 2: Summary of SESAKA Scheme

Section 4.4 is completed. In this phase, data is collected from the consumer nodes to create blocks of transactions and add them to the blockchain.

$SESDAG_1$: The numerous CN in the smart consumer networks use the established session keys in the $SESAKA$ phase to securely send the sensitive energy-related data to the DS. DS creates transactions out of the received data in the format of $Txn_i = \langle ID_D, TS_D, CNRead_i \rangle$. DS uses its private key pr_D to generate $Sign_{Txn_i} = DigSig_{pr_D}(H(Txn_i))$, where signature generation algorithm in "Elliptic Curve Digital Signature Algorithm (ECDSA)" is used for $DigSig(\cdot)$.

$SESDAG_2$: Once DS collects $count_{txn}$ number of transactions, it creates a block as shown in Figure 3. Once the block is created, one of the distribution servers will be elected as the leader. The "Practical Byzantine Fault Tolerance ($PBFT$)" consensus algorithm [12] is executed among the servers in the P2P blockchain server network to achieve an agreement over the veracity of the block. If the block is verified to have correct transactions, signatures, and Merkle tree root by $2(N_D - 1)/3 + 1$ servers in a network of N_D servers, then it is deemed valid and added to the blockchain.

Block Header	
Block Version ($Block\_Version$)	Unique block version number
Previous Block Hash ($Previous\_Block\_Hash$)	Hash value of previous block
Merkle Tree Root ($MTRoot_{TX}$)	Merkle tree root on transactions
Timestamp (TS_D)	Block creation time
Owner of Block	Distribution server (DS)
Public key of transactions verification	Pub_D
Public key of block signer	Pub_D
Block Payload (Encrypted Transactions)	
MV Transactions Txn_i	$\{Txn_i \vert i = 1, 2, \cdots, count_{txn}\}$
ECDSA signature on Block	$DigSig_{Block}$
Current Block Hash ($Current\_Block\_Hash$)	Hash value of current block

Figure 3: Structure of a Block in the blockchain

5 SECURITY ANALYSIS

5.1 Informal Analysis

In the proposed scheme, during the authentication and key agreement phase, CN sends $MSG_{CD1} = \langle TID_C, t_C, A_C, Q_C, A'_C, S'_{2C}, S_{1C}, E_C, TS_C \rangle$ to DS; DS sends $MSG_{CD2} = \langle TID_D, TID_C^*, v_D, B_D, Q_D, B'_D, S'_{2D}, S_{1D}, F_D, TS_D \rangle$ to CN; CN sends $MSG_{CD3} = \langle SKV_{CD}, TS_{CD} \rangle$ to DS.

5.1.1 Replay Attack. Three messages MSG_{CD1}, MSG_{CD2} and MSG_{CD3} use the timestamps TS_C, TS_D, TS_{CD}. Every timestamp is verified at the receiver to check that it does not exceed the justified freshness limit. If this condition is satisfied, the rest of the protocol proceeds. This ensures that any adversary can replay no message, and the proposed $SESAKA$ scheme is resistant to replay attack.

5.1.2 Man-in-the-Middle(MiTM) Attack. Any change to the parameters $t_C, A_C, A'_C, S'_{2C}, S_{1C}, E_C$ during message transit of Msg_{CD1} will fail the signature verification conditions and lead to immediate identification of MiTM attack. Similarly, changes to v_D, B_D, Q_D, B'_D, S'_{2D}, S_{1D} and F_D in Msg_{CD2} and SKV_{CD}, TS_{CD} in Msg_{CD3} will be identified in the verification on the receiver end. This shows that the proposed scheme is resistant to MiTM attacks.

5.1.3 Impersonation Attack. To impersonate CN, Msg_{CD1} needs to be fabricated as $MSG_{CD1}^{Av} = \langle TID_C^{Av}, t_C^{Av}, A_C^{Av}, Q_C^{Av}, A_C'^{Av}, S_{2C}'^{Av}, S_{1C}^{Av}, E_C^{Av}, TS_C^{Av} \rangle$. This requires the adversary Av to generate the A_C, E_C, and TS_C. However, Av is unaware of the secrets a_C, e_C and c_1 and this cannot impersonate CN. Similarly, to impersonate DS, Msg_{CD2} needs to be fabricated as $Msg_{CD2}^{Av} = \langle RID_D^{Av}, TID_C^{*Av}, v_D^{Av}, B_D^{Av}, Q_D^{Av}, B_D'^{Av}, S_{2D}'^{Av}, S_{1D}^{Av}, F_D^{Av}, TS_D^{Av} \rangle$. This requires the adversary Av to generate the B_D, F_D, and TS_D. However, Av is unaware of the secrets b_D, f_D and d_1 and this cannot impersonate DS. Thus, the proposed scheme is resistant to impersonation attacks.

5.1.4 Privileged Insider Attack. The secret credentials ID_C and c_1 used for the registration of CN and ID_D and d_1 used for the registration of DS are pre-loaded into their memory in offline mode and never shared in any public channel throughout the scheme. Hence, the scheme is resistant to privileged insider attacks.

5.1.5 *Ephemeral secret leakage (ESL attack).*
The session key $SK_{DC} = H(DK_{DC} || H(pr_C || e_C || RID_C || TS_C) || H(pr_D || f_D || RID_D || TS_D)) = H(DK_{CD} || H(pr_C || e_C || RID_C || TS_C) || H(pr_D || f_D || RID_D || TS_D)) = SK_{CD}$ uses the long term secrets pr_C, RID_C, RID_D and short term secrets a_C, e_C, b_D and f_D. The compromise of short-term secrets keeps the session key safeguarded due to the security offered by long-term secrets. Similarly, the compromise of long-term secrets keeps the session key safeguarded due to the security offered by short-term secrets.

5.1.6 *Physical Device Node Capture Attack.*
Capture of physical consumer smart node and performing power analysis attacks and timing attacks reveals the parameters $\{(RID_C, TID_C, TC_C), H(\cdot), E_q(a, b), G, (pr_C, Pub_C)\}$ from its memory. However, the exposed credentials do not compromise the communication with the other working consumer smart nodes as none of their credentials are revealed. Hence, the proposed scheme is secure against physical node capture attacks.

5.1.7 *Denial-of-Service (DoS) Attack.*
Once the DS receives the message Msg_{CD1}, it verifies the signature using the two conditions $Q_C \stackrel{?}{=} H(pr_C || e_C || RID_C || TS_C) \cdot G + S_{1C} \cdot E_C$ and $S'_{2C} \cdot A'_C \stackrel{?}{=} H(pr_C || e_C || RID_C || TS_C) \cdot A_C + x'_C \cdot Q_C$. If any of these conditions fail, the authentication process halts, and the consumer smart node is not allowed to access the server any further. Thus, the proposed scheme is resistant to DoS attacks.

5.1.8 *Anonymity and Untraceability.*
The messages MSG_{CD1} and MSG_{CD2} only use the temporary and pseudo-identities of CN and DS. Their true identities ID_C and ID_D are never revealed in any of the public channels. Hence the scheme supports anonymity. In addition, none of the messages contain any parameters that can relate to multiple messages from any entity. Hence, the proposed scheme supports untraceability.

5.1.9 *Privacy Preservation.*
The proposed scheme applies zero-knowledge protocol using $ECDSA$ over the points E_C, F_D and on the second part of signatures by sending S'_{2C} and S'_{2D} instead of the actual sign S'_{2C} and S'_{2D}. Thus, the proposed scheme ensures that the privacy of the entities CN and DS is preserved.

5.2 Formal Verification using AVISPA

The proposed SESAKA scheme is simulated using the " Security Protocol ANimator (SPAN)" simulator in the "Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool with the code written in "High-Level Protocol Specification Language (HLPSL)" that uses temporal logic [8, 9]. The code consists of three roles for the energy supplier, consumer smart node, and distribution server. A secrecy goal on random secrets and secret keys, and authentication goal on timestamps and private variables are applied to verify resistance against MiTM and replay attacks [11, 28–33]. The simulation results of the proposed scheme is shown in Figure 4.

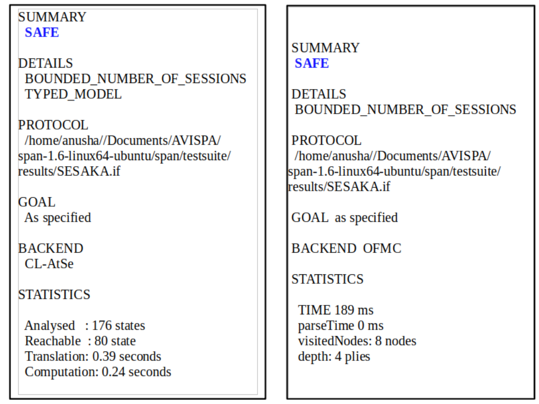

Figure 4: AVISPA simulation results of Proposed Scheme on Cl-ATSe and OFMC backends

6 COMPARATIVE ANALYSIS

6.1 Computation Cost Analysis

The cryptographic operations used in the proposed SESAKA scheme are experimented using the widely-accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [5] which gives the average time of execution of each operation. MIRACL is a cryptographic library with "open source SDK for Elliptic Curve Cryptography" written in "C/C++ programming language".

Table 1: Average execution time (in milliseconds)

Primitive	Average time on Raspberry PI 3 (ms)	Average time on server (ms)
T_h	0.309	0.055
T_{exp}	0.228	0.072
T_{ecm}	2.288	0.674
T_{eca}	0.016	0.002
T_{enc}	0.018	0.001
T_{dec}	0.014	0.001
T_{bp}	32.084	4.603

The notations T_{bp}, T_{exp}, T_{ecm}, T_{eca}, T_{enc}/T_{dec} and T_h are used to denote the execution time needed for a "bilinear pairing ", a "modular exponentiation", an "elliptic curve point (scalar) multiplication", an "elliptic curve point addition", a "symmetric key encryption/decryption using the Advanced Encryption Standard (AES-128) encryption [4]", and a "one-way hash function using the SHA-256 hashing algorithm", respectively.

In the following, we consider two situations for experiments using the MIRACL:
Scenario 1. Under this scenario, we consider the platform under a server having the setting: "Ubuntu 18.04.4 LTS, with memory: 7.7 GiB, processor: Intel Core™ i7-8565U CPU @ 1.80GHz × 8, OS type: 64-bit and disk: 966.1 GB". Table 1 shows the maximum, minimum and average run-time (in milliseconds) for each cryptographic primitive for 100 runs.

Scenario 2. Under this scenario, we consider the platform for an IoT smart device or a user mobile device using the setting: "Raspberry PI 3 B+ Rev 1.3, with CPU: 64-bit, Processor: 1.4 GHz Quad-core, 4 cores, Memory (RAM): 1GB, and OS: Ubuntu 20.04 LTS, 64-bit" [6]. Table 1 shows the maximum, minimum and average run-time (in milliseconds) for 100 runs.
Table 3 shows the computation cost of each of the compared schemes along with the execution time in ms using the above MIRACL library.

Table 2: Comparison of computational costs

Schemes	Consumer Smart Node / Smart Device	Server
Qi and Chen [24]	$4T_h + 4T_{ecm} + T_{eca} \approx$ 10.404ms	$4T_h + 4T_{ecm} + 2T_{eca}$ $\approx$ 2.918 ms
Sadhukhan *et al.* [25]	$4T_h + 8T_{ecm} + 2T_{eca} + T_{enc} \approx$ 19.59 ms	$4T_h + 8T_{ecm} + 2T_{eca} + T_{enc} \approx$5.617 ms
Xiang *et al.* [34]	$8T_h + 4T_{ecm} + 2\,T_{eca} + 5T_{enc} + 2T_{bp} + T_{dec}$ $\approx$ 75.928 ms	$8T_h + 4T_{ecm} + 2\,T_{eca} + 5T_{enc} + 2T_{bp} + T_{dec}$ $\approx$ 12.342 ms
Agilandeeswari *et al.* [7]	$6T_h \approx$ 1.854 ms	$6T_h$ 0.33 ms
SESAKA	$7T_h + 12T_{ecm} + T_{eca}$ $\approx$ 29.635 ms	$9T_h + 12T_{ecm} + T_{eca}$ $\approx$ 8.585 ms

NA: Not Applicable

6.2 Communication Cost Analysis

The identities and random secrets are taken to be 160 bits each. The length of output of hash function, the ciphertext block of "symmetric key encryption/decryption using AES-128 encryption [4]" are taken as 256 bits and 128 bits. A point $P = (x_P, y_P)$ on the elliptic curve $E_q(a, b)$ is taken as (160 + 160) = 320 bits, with the coordinates x_P and y_P considered as 160 bits each, assuming that 160-bit ECC provides the same security level as that for 1024-bit RSA public key cryptosystem [10]. Moreover, the timestamp is taken as 32 bits.

Table 3: Comparison of communication overheads

Schemes	Total messages	Total cost (in bits)
Qi and Chen [24]	2	1216
Sadhukhan *et al.* [25]	3	2048
Xiang *et al.* [34]	6	4576
Agilandeeswari *et al.* [7]	3	1856
SESAKA	3	4960

6.3 Security Features Analysis

The proposed scheme is compared with the schemes [7, 24, 25, 34] to verify the suppoerted security features. It can be understood that even though the schemes [7, 24, 25] have low computation and communication cost, they do not support many essential security features of anonimity, untraceability and blockchain support. Even though the scheme [34] supports some of the required features, it has very high computation cost.

Table 4: Comparison of security and functionality features

Features	Qi and Chen [24]	Sadhukhan [25]	Xiang *et al.* [34]	Agilandeeswari *et al.* [7]	SESAKA
$\mathcal{F}_1$	×	×	✓	×	✓
$\mathcal{F}_2$	×	×	✓	×	✓
$\mathcal{F}_3$	✓	×	✓	×	✓
$\mathcal{F}_4$	×	×	×	×	✓
$\mathcal{F}_5$	×	×	×	×	✓
$\mathcal{F}_6$	×	✓	✓	×	✓
$\mathcal{F}_7$	✓	✓	✓	✓	✓
$\mathcal{F}_8$	✓	✓	✓	×	✓
$\mathcal{F}_9$	✓	✓	✓	✓	✓
$\mathcal{F}_{10}$	✓	✓	✓	×	✓
$\mathcal{F}_{11}$	×	✓	×	×	✓
$\mathcal{F}_{12}$	×	×	✓	×	✓
$\mathcal{F}_{13}$	✓	×	✓	×	✓
$\mathcal{F}_{14}$	×	×	×	×	✓

Note: $\mathcal{F}_1$: "anonymity"; $\mathcal{F}_2$: "untraceability"; $\mathcal{F}_3$: "user impersonation attack"; $\mathcal{F}_4$: "physical node capture attack"; $\mathcal{F}_5$: "ephemeral secret leakage (ESL) attack"; $\mathcal{F}_6$: "privileged insider attack"; $\mathcal{F}_7$: "replay attack"; $\mathcal{F}_8$: "man-in-the-middle attack"; $\mathcal{F}_9$: "mutual authentication"; $\mathcal{F}_{10}$: "unauthorized login detection"; $\mathcal{F}_{11}$: "Denial-of-Service (DoS) attack"; $\mathcal{F}_{12}$: "offline guessing attacks"; $\mathcal{F}_{13}$: "privacy preservation"; $\mathcal{F}_{14}$: "blockchain support"; N/A: "not applicable in a scheme"; ✓: "a feature is supported in a scheme or resistant against the specified attack"; ×: "a feature is not supported in a scheme or it is not resilient against the specified attack"

7 BLOCKCHAIN SIMULATION

One of the distribution servers in the P2P network is elected as the leader in a round-robin fashion and proposes a block. All the servers apply the PBFT consensus [12] to verify the block. If the verification is successful, the block is added to the chain.
The blockchain simulations were performed on a platform having the environment: "Ubuntu 18.04.4 LTS, 64-bit OS with Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 4 GB RAM" using Node.js language with VS CODE 2019 [18]. The purpose of this simulation is to study the effect of increase in the number of distribution servers, transactions and blocks over the computational time.
The following three scenarios are considered.
Scenario 1: The first scenario tests variation in the number of transactions per block keeping the number of P2P nodes and the number of blocks mines fixed at 11 and 31, respectively. The computational time is shown in Fig. 5(a).
Scenario 2: The second scenario tests the variation of the total number of blocks mined keeping the number of P2P nodes and the number of transactions per blocks fixed to 11 and 32, respectively. The computational time is shown in Fig. 5(b).
Scenario 3: The third scenario tests the variation in the number of P2P nodes participating in the mining process while keeping the number of blocks mined and the number of transactions per block fixed at 10 and 25, respectively, as shown in Fig. 5(c).

8 CONCLUSION

A novel ECC-based authentication scheme SESAKA has been proposed for smart energy systems (SES) that preserves privacy using zero knowledge protocol. The data from the SES is sent to a P2P blockhain based distribution server network which performs data aggregation, creates blocks of transactions out of sensor data and performs consensus before storing the data on the chain. The

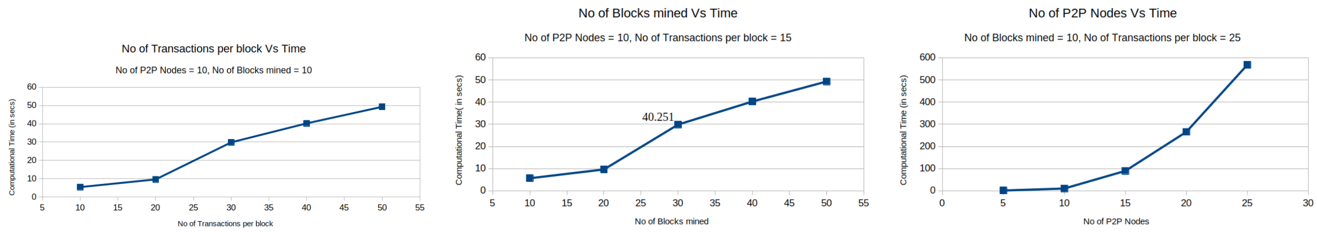

Figure 5: Blockchain simulation results: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3

scheme is analysed using informal analysis and formal security verification using AVISPA tool. Comparative analysis with similar scheme in smart grids show that SESAKA achieves critical security features with reasonable computation and communication cost while preseving privacy and supporting blockchain technology.

REFERENCES

[1] Bad Europe's Energy Crisis. (2022). https://foreignpolicy.com/2022/08/26/europe-energy-crisis-natural-gas-economy-winter/

[2] China's Power Crisis Conundrum. (2022). https://www.scmp.com/economy/china-economy/article/3190313/chinas-power-crisis-why-it-happening-and-what-does-it-mean

[3] Europe's Energy Crisis Conundrum. (2022). https://www.iss.europa.eu/content/europes-energy-crisis-conundrum

[4] Advanced Encryption Standard. (2001). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce. Accessed on June 2020.

[5] MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library. (2020). https://github.com/miracl/MIRACL Accessed on October 2020.

[6] Raspberry Pi 3 Model B+. (2020). https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/ Accessed on April 2020.

[7] L Agilandeeswari, Swapnil Paliwal, Anvita Chandrakar, and M Prabukumar. 2022. A New Lightweight Conditional Privacy Preserving Authentication and Key Agreement Protocol in Social Internet of Things for Vehicle to Smart Grid nNtworks. *Multimedia Tools and Applications* (2022), 1–28.

[8] AVISPA. 2020. Automated Validation of Internet Security Protocols and Applications. (2020). http://www.avispa-project.org/. Accessed on October 2020.

[9] AVISPA. 2020. SPAN, the Security Protocol ANimator for AVISPA. (2020). http://www.avispa-project.org/. Accessed on October 2020.

[10] E. Barker. 2016. Recommendation for Key Management. (2016). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf Special Publication 800-57 Part 1 Rev. 4, NIST, 01/2016. Accessed on July 2020.

[11] Basudeb Bera, Anusha Vangala, Ashok Kumar Das, Pascal Lorenz, and Muhammad Khurram Khan. 2022. Private Blockchain-envisioned Drones-assisted Authentication Scheme in IoT-enabled Agricultural Environment. *Computer Standards & Interfaces* 80 (2022), 103567.

[12] M. Castro and B. Liskov. 2002. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.* 20, 4 (2002), 398–461.

[13] Michael R. Clark and Kenneth M. Hopkinson. 2013. Towards an understanding of the tradeoffs in adversary models of smart grid privacy protocols. In *2013 IEEE Power & Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESMG.2013.6672334

[14] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1989. The Knowledge Complexity of Interactive Proof-Systems. *SIAM Journal on computing* 18, 1 (1989), 186–208.

[15] Shafi Goldwasser, Silvio Micali, and Chales Rackoff. 2019. *The Knowledge Complexity of Interactive Proof-Systems*. Association for Computing Machinery, New York, NY, USA, 203âĂŞ225. https://doi.org/10.1145/3335741.3335750

[16] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2019. Differential Privacy Techniques for Cyber Physical Systems: A Survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 746–789.

[17] D. Johnson, A. Menezes, and S. Vanstone. 2001. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security* 1, 1 (2001), 36–63.

[18] Kashish Khullar. 2019. Implementing PBFT in Blockchain). (2019). https://medium.com/coinmonks/implementing-pbft-in-blockchain-12368c6c9548 Accessed on August 2022.

[19] J Andrew Onesimu, J Karthikeyan, and Yuichi Sei. 2021. An Efficient Clustering-based Anonymization scheme for Privacy-preserving Data Collection in IoT based Healthcare Services. *Peer-to-Peer Networking and Applications* 14, 3 (2021), 1629–1649.

[20] Kisung Park, Joonyoung Lee, Ashok Kumar Das, and Youngho Park. 2022. BPPS:Blockchain-Enabled Privacy-Preserving Scheme for Demand-Response Management in Smart Grid Environments. *IEEE Transactions on Dependable and Secure Computing* (2022). https://doi.org/10.1109/TDSC.2022.3163138

[21] Shruti Patil, Shashank Joshi, and Deepali Patil. 2020. Enhanced Privacy Preservation using Anonymization in IoT-enabled Smart hHmes. In *Smart Intelligent Computing and Applications*. Springer, 439–454.

[22] Andrew Paverd, Andrew Martin, and Ian Brown. 2014. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. *Tech. Rep* (2014).

[23] Andrew Paverd, Andrew Martin, and Ian Brown. 2014. Security and Privacy in Smart Grid Demand Response Systems. In *Smart Grid Security*. Springer International Publishing, Cham, 1–15. https://doi.org/10.1007/978-3-319-10329-7_1

[24] Mingping Qi and Jianhua Chen. 2020. Two-pass Privacy Preserving Authenticated Key Agreement scheme for Smart Grid. *IEEE Systems Journal* 15, 3 (2020), 3201–3207.

[25] Dipanwita Sadhukhan, Sangram Ray, Mohammad S. Obaidat, and Mou Dasgupta. 2021. A Secure and Privacy Preserving Lightweight Authentication Scheme for Smart-Grid Communication using Elliptic Curve Cryptography. *Journal of Systems Architecture* 114 (2021), 101938.

[26] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. 1987. Non-Interactive Zero-Knowledge Proof Systems. In *Conference on the Theory and Application of Cryptographic Techniques*. Springer, 52–72.

[27] Parminder Singh, Mehedi Masud, M. Shamim Hossain, and Avinash Kaur. 2021. Blockchain and Homomorphic Encryption-based Privacy-preserving Data Aggregation model in Smart Grid. *Computers & Electrical Engineering* 93 (2021), 107209.

[28] Anusha Vangala, Basudeb Bera, Saurav Saha, Ashok Kumar Das, Neeraj Kumar, and YoungHo Park. 2021. Blockchain-Enabled Certificate-Based Authentication for Vehicle Accident Detection and Notification in Intelligent Transportation Systems. *IEEE Sensors Journal* 21, 14 (July 2021), 15824–15838.

[29] Anusha Vangala, Ashok Kumar Das, Vinay Chamola, Valery Korotaev, and Joel J.P.C. Rodrigues. 2022. Security in IoT-enabled Smart Agriculture: Architecture, Security Solutions and Challenges. *Cluster Computing* (April 2022).

[30] Anusha Vangala, Ashok Kumar Das, Neeraj Kumar, and Mamoun Alazab. 2021. Smart Secure Sensing for IoT-Based Agriculture: Blockchain Perspective. *IEEE Sensors Journal* 21, 16 (August 2021), 17591–17607.

[31] Anusha Vangala, Ashok Kumar Das, and Jong Hyuk Lee. 2021. Provably-secure Signature-based Anonymous User Authentication protocol in an IoT-enabled Intelligent Precision Agricultural environment. *Concurrency and Computation: Practice and Experience* (March 2021), e6187.

[32] Anusha Vangala, Ashok Kumar Das, YoungHo Park, and Sajjad Shaukat Jamal. 2022. Blockchain-Based Robust Data Security Scheme in IoT-Enabled Smart Home-Envisioned Ubiquitous Computing Environment. *Computers, Materials & Continua* 72, 2 (March 2022), 3549–3570.

[33] Anusha Vangala, Anil Kumar Sutrala, Ashok Kumar Das, and Minho Jo. 2021. Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming. *IEEE Internet of Things Journal* 8, 13 (2021), 10792–10806.

[34] Xinyin Xiang and Jin Cao. 2022. An Efficient Authenticated Key Agreement Scheme supporting Privacy-Preservation for Smart Grid Communication. *Electric Power Systems Research* 203 (2022), 107630.

[35] Liehuang Zhu, Meng Li, Zijian Zhang, Chang Xu, Ruonan Zhang, Xiaojiang Du, and Nadra Guizani. 2019. Privacy-Preserving Authentication and Data Aggregation for Fog-Based Smart Grid. *IEEE Communications Magazine* 57, 6 (2019), 80–85.