IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 3, 2020, accepted January 18, 2020, date of publication January 22, 2020, date of current version February 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968621

Privacy Enhancement Scheme (PES) in a Blockchain-Edge Computing Environment

BONNAH ERNEST AND JU SHIGUANG
School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
Corresponding author: Ju Shiguang (jushig@ujs.edu.cn)

This work was supported in part by the National Key Research and Development Program, China, under Grant 2016YFD0702001, and in part by the Modern Agriculture Projects of Jiangsu Province under Grant BE201735B.

ABSTRACT With the public key being the parameter users are mostly addressed by on blockchain network, an intruder can connect transactional patterns to the public key and make a probable revelation of the identity of the user. Due to the diversity in approaches in achieving privacy, integrating the principle of transparency in any blockchain – edge computing platform will present some structural security challenges. Thus, an attempt should be made to achieve confidentiality whilst not eliminating a key principle of blockchain – transparency. Based on elliptic curve cryptosystem (ECC), we present a privacy-aware scheme that preserves the privacy details of the user in a blockchain–edge computing environment based on a combination of randomly generated public keys and digital signatures. The resilience and practicality of our scheme were tested on AVISPA and NS2 respectively. The results indicated our scheme was robust against attacks, efficient and low on computation resources of edge devices.

INDEX TERMS Privacy, public key, blockchain, location context, edge computing paradigm.

I. INTRODUCTION

The increase in the number of Internet of Things (IoT) devices as well as their demands, has revealed the shortfalls and bottlenecks in traditional cloud computing networks. The many challenges such as bandwidth constraints, latency, and jitter pose major limitations in the provision of delay-sensitive application services. The emergence of edge computing which brings storage and processing resources to the edge of the network in proximity to the end user has in recent years being identified by the research community to be the tool to fill in the gaps identified in cloud computing. Over the years, users have migrated from the consumption of data at the edge of the network to becoming primary producers of data that require huge data processing capabilities [1]. The generated data by users at the edge of the network does not only require big data processing capabilities but also network connectivity as well as storage resources as the data are processed and presented to users [2]. Complementing cloud computing networks with the edge network infrastructure presents us with the opportunity to meet the growing demand of sophisticated edge users. Real-time applications due to their proximity to cloud-like computing resources at the edge of the network will have mobility support, context awareness, low latency and knowledge of their local network conditions [3]. Users can now access information relative to their location, health details, purchasing preferences, etc. in a real-time with no preexisting limitation as seen in traditional cloud computing [4].

The continuous progress in the delivery of services with little or no bottlenecks at the edge of the network introduces the need to secure these devices against any form of intruder attack and manipulation. The nature of the landscape in edge infrastructures makes the network susceptible to all forms of attacks in the absence of an efficient security framework. Most often than not, no single individual or entity has complete control over the entire network. Services are mostly outsourced to third parties. Virtualization which is a key component of the various edge computing paradigms is an opportunity for any adversary to exploit the network. Since different service providers right from the network core, to the edge infrastructure and the virtualization infrastructure, control various aspects of the network, it is highly difficult to predict where any attack on the network is going to emanate from. Users' privacy can be compromised by adversaries from edge data centers,

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

infrastructure providers, service providers, service providers as well as users [1]. Security in any network infrastructure should therefore not be compromised no matter the number of elements involved in service delivery. Privacy relative to users' credentials as well as information exchanged over an edge network has, therefore, come to the fore since costumers want the assurance from service providers that their sensitive information stored or processed on remote servers will not be accessed and manipulated by any malicious intruder.

Most privacy schemes are designed with the aim of protecting data against external adversaries. Nonetheless, the scope of attacks can emanate from internal adversaries who are already authorized and authenticated to perform certain functions within the network setup. Authorized users may either be honest but curious, malicious themselves or could lose their device due to theft or loss. Authorized devices can also be hacked into by sophisticated intruders. This places these internal intruders in a position to access all the exchanged information that traverses the edge network. Thus, an internal adversary with enough access rights can alter information flow and even inject false information into the network [5]. With context and location awareness enabled within the edge computing paradigms, the credentials of a user can be extracted from data exchanged with other elements in the network in the absence of an efficient security mechanism [6]. This is because location information of users can be easily perceived as users are likely to access the services of a particular service provider or edge server over a period. Once both external and internal adversaries gain access to the sensitive information traversing any edge data center, the associated dangers cannot be quantified. For example, in cyber-physical systems such as an electric power generation and delivery setup, smart meters which store privacy information of users and their respective households are deployed in the homes. In the hands of an attacker, these sensitive details of various households can be retrieved from the smart meters leading to privacy leakage [1]. Hence it is very important to protect a given user's identity information, policy components (rules) during any form of integration of blockchain into edge computing as well as transactions exchanged between nodes on the platform [7].

The major contribution of the paper are listed below:

1. We discuss the limitation of using only encryption as a means of achieving privacy in a blockchain – edge computing environment since this approach is not comprehensive enough.
2. We then proceed to introduce PES which is based on randomly generated public keys based on the elliptic curve cryptosystem to achieve user untraceability and anonymity as well as a digital signature which is computed based on messaged content.
3. Since encryption does not make transmitted messages tamperproof, we integrated indexing of every transaction into our schemes to ensure the messages that are either captured, replayed or tampered with by an adversary are identified. Previous transactions exchanged between were hashed and the output used as indexes.

The remainder of the paper is organized as follows: Section II presents the current trends of achieving privacy in a blockchain – edge computing environment while discussing the limitations. Our proposed scheme PES is itroduced in Section III with cryptanalysis on the scheme in Section IV. AVISPA is used to perform further tests on our scheme in Section V with practical simulations of the scheme using NS2 in Section VI. A comparison of the performance of our scheme with an earlier proposed scheme is done in Section VII with the conclusion in Section VIII.

II. PRIVACY IN BLOCKCHAIN-EDGE COMPUTING

The principles of blockchain technology have been introduced into the realm of computing to enhance the many shortfalls identified in cloud computing and other related fields such as IoT. Bitcoin, a cryptocurrency system, which was built on the principles of blockchain has been used to provide security and privacy in peer-to-peer networks with topologies similar to IoT [8]. The decentralized approach of blockchain eliminates the many bottlenecks and points of failures in centralized networks. With the elimination of the point of failure comes the decrease in latency in a typical client-server network. Furthermore, decentralized networks scale better than the one with central control.

Inherently, blockchain is designed to protect the identities of users in the networks. Users within a blockchain whether it is private or public are addressable on the network by their public keys. In some deployment of blockchain, the public key which is used to address the users is hashed to conceal it from any malicious outsider [9]. This definitely may go a long way to achieve user anonymity within the network. However, the principle of transparency in the blockchain introduces some complexity and complication with the integration of this technology in edge computing. All transactions on a blockchain are transpired in the open. Nodes take delivery of a transaction, validate the transaction and then forward the transaction to all its neighboring nodes. Thus, after a period every node will have access to the content of the transaction. Every blockchain has rules governing its operation since nodes in the blockchain do not necessarily have to trust each other. The rules for every blockchain are embedded into the blockchain application which the nodes sign onto. Nodes validate each transaction by confirming that each node complies with all the rules specified before forwarding them on to the neighboring nodes. Maintaining privacy relative to sensitive information kept between two parties become complicated once all the nodes in the network have access to validate the transaction. To resolve this challenge, sensitive portions of transactions can, therefore, be encrypted with the public key of the intended recipient by the sender of the transaction. The receiver then can decrypt the received transaction using its secret key. This will prevent malicious intruders from having access to this sensitive user information. However, the reality still exists that honest but curious users (nodes)

who are already authorized to be part of the network can make inferences by tracking the activities of a user's public key. A connection can be made by tracking payments and IP addresses leading to the identity of the user being revealed [10], [11]. Therefore, the highest level of privacy must be guaranteed not only against external attackers but internal adversaries who may want to leverage on the access rights within the network to gain access to sensitive information traversing the network.

Not much work has been done to enhance privacy in any of the edge paradigm environment integrated with blockchain. To the best of our knowledge, this will be the first paper to deal with privacy in a blockchain-edge computing environment without compromising transparency. The focus of our paper is to design a scheme that focuses on privacy in layers above the core of the network infrastructure within a blockchain integrated edge computing environment. Our scheme seeks to ensure that the privacy of a user i.e. his credentials and information exchanged with users are secured within a blockchain – edge platform while not compromising on fundamental principles underpinning blockchain.

A. LIMITATION OF CURRENT TREND IN BLOCKCHAIN-EDGE COMPUTING ENVIRONMENT

Various forms of encryption techniques have been employed in an attempt to maintain the confidentiality of transactions exchanged between elements in a blockchain - edge computing environment. Over the years, efforts have been made to improve encryption schemes to make them more robust and comprehensive to withstand any attack from external adversaries. Identity-based encryption (IBE) [12], Attribute-based encryption (ABE) [13], Proxy re-encryption (PRE) [14] and variants of Homomorphic encryptions are the main encryption mechanisms that have been enhanced in recent times to achieve data confidentiality.

The deployment IBE variants allow any pair of users to communicate and verify each other's signatures without the exchange or pre-distribution of a set of pairwise keys. Users generate known unique identifiers or arbitrary string that exclusively identifies them. Users on the network become addressable on the network through these generated known identifiers. With the known identifiers playing the role of public keys within IBE framework, the private keys of respective users are computed and generated using a trusted third party or IBE server such as a Private Key Generator. Despite the aim of using IBE to address the computation-intensive limitations of public-key cryptography in resource-constrained devices, the approach has been challenged by two main drawbacks. The involvement of a trusted third party or IBE server introduces an element of centralization in the setup. Issues of a single point of failure, as well as scalability, will arise in such environments. In order to keep the privacy of the private keys generated by the trusted third parties or the IBE server, a secure channel is required between the user and the trusted third party or the IBE server. However, in real-time networks, it is difficult to establish a completely secured network between the two entities.

Taking encryption a step further from IBE, ABE schemes which are based on the concept of public key cryptography encrypts information based on either a set of attributes (e.g. roles, location, etc.), subsets of attributes or policies defined based on a set of attributes. Users with the corresponding matching attributes issued by a trusted third party are the only ones who can then decrypt the ciphertext. Threshold-based strategies on attributes are supported in ABE schemes. What this means is decryption operation can be executed only when the matching elements between the attributes of users and ciphertexts reach the threshold value. Various variants of ABE schemes have been proposed in recent times. These include Key Policy attribute based encryption (KP-ABE), Ciphertext policy attribute based encryption (CP-ABE), Hierarchical attribute-based encryption (HABE), Multi-level attribute-based encryption (MABE) and Attribute-based Encryption scheme with Non-Monotonic Access Structures [1]–[15]. The limitations of each variant scheme of ABE have been extensively discussed by Lakshmi *et al.* [15]. Aside the presence of a trusted third party, all the variants of ABE schemes have fundamental limitations that affect the core of the security and access control services intended to be provided.

PRE schemes are cryptosystems that allow a third party designated as a semi-trusted proxy to alter a ciphertext that has been encrypted by data owner so that it can be decrypted by the intended recipient or user. The semi-trusted proxy alters the already encrypted data of the owner through the use of a re-encryption key. There are two major limitations of PRE schemes namely [16]: Firstly, the semi-trusted proxy can divert the ciphertext of the data owner and reverse the process the ciphertext based on computational properties of discrete logarithm without getting permission from either the data owner and the intended recipient. Again, the semi-trusted proxy together with the data owner can collude to know the private key of the intended recipient. This will make it possible for the proxy to decrypt the re-encrypted ciphertext to gain access to the original message. In recent times, schemes have been proposed to solve the identified limitations of conventional PRE [17]. However, most of these schemes dealt with the bidirectional challenge of PRE. Not much work has been done to improve the limitation posed by the possibility of collusion between the data owner and the semi-trusted proxy to unveil the secret key of another user.

Homomorphic encryption is an encryption mechanism that allows for computation to be performed on ciphertext as if it was a plaintext with the use of a decryption key [18]. The computations can be done without any compromise on the encrypted data. Among all the encryption techniques, homomorphic encryption is known to be the best among all the schemes in achieving transactional privacy in the desired environment [9], [19]. Nonetheless, this scheme among other schemes has its limitations. For a comprehensive deployment within an environment such as ours, further developments

in the encryption scheme have to be undertaken to make them suitable for deployment in our desired environment. With homomorphic encryption, once a user sends encrypted information into the data center, it cannot be either indexed or searched [20]. However, these are some of the operations users will perform on data sets.

Most importantly, the underlying crypto-strength of homomorphic encryption has in recent analysis been questioned. Like many other schemes, this scheme is seen to leak private information of its users [21]. Thus, encryption is not sufficient to achieve privacy in a blockchain- edge computing network. This attempt to secure user privacy and anonymity must be complemented by additional mechanisms to make any network security as robust as possible. Nonetheless, these complementary schemes should take into consideration the computational and limitations of the devices at the edge of the network.

III. OUR PROPOSED SCHEME

With the integration of blockchain into edge computing paradigms, both privacy preservation is achieved through the use of encryption schemes and addressing users through their respective public keys. However, both means have been seen to have limitations and not sustainable enough, thus, posing a significant threat to the privacy of users. Our scheme focused on eliminating the challenge related to the public keys whilst complementing the encryption mechanism to enhance user privacy, anonymity and untraceability.

Based on elliptic curve cryptosystem (ECC), we present a privacy-aware scheme that preserves the privacy details of the user in a blockchain–edge computing environment based on the random generation of public keys and digital signatures. ECC provides the smallest key size per equivalent strength of any traditional public-key cryptosystem [22]. The use of lightweight ECC in a flexible decentralized blockchain-edge computing environment, presents an opportunity for public keys and signatures to be generated by both resource-constrained and computationally powerful nodes without the involvement of any centralized entity with a knock-on effect on data confidentiality, authentication and integrity. Public keys and digital signatures can therefore be efficiently generated within the desired environment with less computational resource and time.

To protect the anonymity of users and achieve untraceability, new public keys will be generated randomly for every transaction exchanged between users based on Public Key Random Generation (PKRG). PKRG is a simple generation mechanism involving the secret key and random nonces. Public keys are expired once the expiration time, λ elapses. PKRG will be called whenever a transaction has to be initiated. The algorithm goes on to generate a new public key if λ elapses. Digital Signature Generation (DSG) is called once PKRG has returned a value (i.e. a valid public key). The values used in the indexation of transactions are generated as part of the DSG process. Since encryption keeps intruders from accessing the contents of the transaction and not necessarily manipulating them, every transaction is indexed by the hashed output transaction to be exchanged as well as the previous transaction between the two users. This is to enable the parties involved to know when a particular message is captured, altered or even replayed by an adversary.

A. AUTHENTICATION IN PES

Though authentication is not the primary goal of our scheme, any efficient privacy enhancement scheme will have a ripple effect on authentication. Authentication in most conventional edge computing environments is based on three factors namely: password, smartcard and biometric characteristics [23]. Two-factor authentication scheme which employs password and smartcard as deployed in [24] allows for exploitation by an intruder due to identified drawbacks [1], [23]. In three-factor authentication scheme, a third security feature of biometric details is added to make scheme more robust.

PES requires a user to first authenticate himself on his mobile device through a 3-factor authentication method using his secret key, password and biometric input (e.g. fingerprint). Mutual authentication between users in a blockchain-edge computing environment departs from the conventional centralized methods of authentication in centralized networks involving a certification authority or trusted third party [2], [25]. Nodes are required to register their identity with the blockchain using their key-pair. The registered identity is a composition of hashes of several identity related attributed [26]. In our case, the registered identity is made up of the public keys, chosen identities, and digital signatures.

Characteristic of any blockchain platform, users are not required to trust each other. However, enhancing authentication definitely enforces trust in any system [27]. Based on the information earlier registered on the blockchain system, nodes authenticate or validate each other. This process does not involve only the sender and the intended recipient but every other node on the network who act as sponsors of the registered information. The use of indexes provides an additional authentication feature. In addition to stamping every transaction with hash output of the transaction, the hash output previous transaction exchanged between the sender and receiver is also referenced as part of the index. Hence, the intended recipient gets to verify the validity of the user by confirming the public key, digital signature as well as the chosen ID. The index allows nodes to know of any tampering, replay and deletion of received transaction from the legitimate sender.

The transaction exchanged between the users will have four sections namely:

- Sender: This will be the public key of the sender of the transaction
- Recipient: This section will contain the public key of the recipient
- Message (m): The message could be a token, request, etc. Together with the generated signature and the identity of the sender, the message will be encrypted with the

TABLE 1. Notation used.

Symbol	Description
U_i, U_j	Users i and j respectively
ID_i, ID_j	Chosen identities of U_i and U_j respectively
S_i, P_i	Secret and Public keys of U_i respectively
f_i	Biometric fingerprint of U_i
Sig_i, Sig_j	Digital signatures of U_i and U_j respectively
$x, y, z \in \mathbb{Z}_p$	Random numbers
$\|$	Concatenation function $(x \| y)$
$H_n(.)$	One way hash function $n = (1,2,3,4)$
G	Additive cyclic group
e	A pairing function
$+$	Point addition operation in elliptic curve group
$\oplus$	Bitwise XOR function $(a \oplus b)$
P	Generator of G in ECC
λ	Public key expiration time
T	Transaction file
$\{F, D\}$	Elements of a digital signature

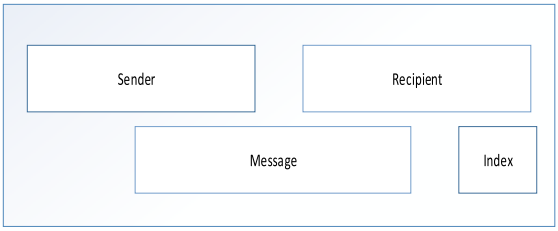

FIGURE 1. Structure of the transaction file.

public key of the intended recipient. Thus, message, m in a scenario U_i sends a transaction or request to U_j will be configured as $E_{P_j}[ID_i \| Sig_{U_i} \| m]$

- Index (I_t) : This is for the sequencing of the messages.

B. ASSUMPTIONS

PES scheme runs on the following assumptions:

1. Every node on the network is enabled as a PES client which operates on principles of blockchain and edge computing.
2. Since blockchain does not support any form of centralization, no trusted third party or certificate authority was involved in the issuance of the private and public keys.
3. As in a blockchain network, all users (nodes) on the network keep a distributed ledger that contains all the identities and corresponding public keys of all other users on the networks.
4. Based on the public parameters, user U_i computes its secret key S_i.
5. Neither internal nor external adversaries have access to any user's credentials such as secret key S_i, and password PW_i which are securely saved in the memory of the mobile device.

Let G_1 and G_2 be a cyclic additive group and multiplicative group respectively both of a large prime order p. We declare e as an admissible pairing function $e : G_1 \times G_1 \rightarrow G_2$ as a map with the following properties [28]:

1. Bilinearity: $e(xP, yQ) = e(P, Q)^{xy}$ for all $P, Q \in G_1$ for all $x, y, z \in Z_p$.
2. Non-degeneracy: There exists $P \in G_1$ such that $e(P, P) \neq 1$.
3. Computability: There is an algorithm to compute $e(P, Q)$ for any $P, Q \in G_1$.

Once a node is enabled as a client, it receives all public parameters needed in the computation of its secret key, generation of its random public keys, etc. These parameters are computed based on the elliptic curve cryptosystem (ECC). The parameters include $\{H_1, H_2, H_3, H_4, P, Z_p, e, G, p\}$.

C. DESCRIPTION OF PES

The PES algorithm is made up of two phases namely Digital Signature Generation (DSG) and Public Key Random Generation (PKRG). The generation of the digital signature and public keys were computed based on the elliptic curve cryptosystem.

Let $P = (x_p, y_p)$ be a point on the elliptic curve $E : y^2 = x^3 + ax + b$ over a prime finite field F_p defined by a and b where $a, b, x_p, y_p \in F_p$. Assume Pis the generator of additive cyclic group, G of prime order p and every point generated on E. If p is an odd prime comprising of a set of integers $\{1, 2, \ldots, p-1\}$, then the following statements are valid [29], [30]:

1) *If $\boldsymbol{p>3}$ then a and b satisfy the equation $\boldsymbol{4a^3+27b^2 \not\equiv 0}$ $\boldsymbol{(mod\ p)}$ and every generated point by P on E shall satisfy the equation $\boldsymbol{y^2=x^3+ax+b}$ except when $\boldsymbol{P = 0}$.*
2) *If $a, b \in F_p$, then $a + b = r$,where r the remainder when $a + b$ is divided by $\boldsymbol{p}$ and $\boldsymbol{0 \leq r \leq p-1}$.*
3) *If $\boldsymbol{a, b \in F_p}$, then $\boldsymbol{a.b = s}$, where s is the remainder when a.b is divided by p and $\boldsymbol{0 \leq s \leq p-1}$.*
4) *If a is a non-zero element in F_p, the inverse of $\boldsymbol{a}$ and $\boldsymbol{p}$, denoted $\boldsymbol{a^{-1}}$, is the unique integer $c \in F_p$ for which $\boldsymbol{a.c = 1}$.*

D. PUBLIC KEY RANDOM GENERATION (PKRG)

PKRG has three functionality blocks namely ***Initiate***, ***Generate*** and ***Validate***.

Initiate: The ***Initiate*** is initialized whenever a transaction file has to be composed. ***Initiate*** takes as input secret key S_i, password PW_i and biometric input f_i). Once the given credentials are correct, ***Initiate*** verifies the validity of the current public key P_i to be used in the yet to be composed transaction. ***Initiate*** does this by checking if the predefined public key expiration time, λ has elapsed. Thus,

$$\textbf{\textit{Initiate}}\ (S_i, PW_i, f_i) \longrightarrow (P_i\ \textbf{\textit{Generate}})$$

Generate: The Public Key Random Generation algorithm ***Generate*** is only initialized when the public key expiration time, λ has elapsed and P_i is invalid.

Validate: The principal idea of the scheme is to have randomly generated public keys. Thus, no two consecutive public keys should be the same. ***Validate*** takes as input the expired public key, P_i and compares it to the newly generated P_i^* returned by ***Validate***. If they are the same,***Validate*** returns invalid and takes the process back to ***Initiate*****,** else valid. Thus, **Validate** $(\mathrm{P_i}) \rightarrow$ Valid/Invalid.

Given the secret key of U_i with identity ID_i as S_i, U_i selects a random nonce $x, y, z \in \mathbb{Z}_p$ and computes the initial public key, P_i:

$$P_i = (S_i)^x \, G \, mod \, p$$

Together with the chosen identity ID_i, U_i shares the initially generated public key P_i with all other nodes on the network. Upon broadcasting his public key, the user will be addressable on the network by the other users through the public key. The chosen identity will be used by users to confirm the identity of the user upon receiving a new public key from U_i likewise other users. ***Initiate***: Every generated public key has an expiration period denoted as λ. Whenever U_i wants to initiate any request or transaction with any user, ***Initiate***: will be initialized to verify the validity of P_i. Given $\lambda = 0$, the PKRG algorithm operates as follows:

1. Given a security parameter $t \in \mathbb{Z}_p$, G is triggered on an input of t to generate G_1 and G_2 of the order p, a generator P in G_1 and an admissible pairing function e.
2. U_i then randomly selects a nonce $\{x, y, z\} \in \mathbb{Z}_p$
3. (G, p, x) are then forwarded to the next phase.

Generate: Given λ has expired rendering P_i invalid,

1. Compute $P_i^* = (S_i)^x \, G \, mod \, p$

Validate: Given the expired public key, P_i and generated public key $\mathrm{P_i^*}$,

1. Confirm if $P_i \neq P_i^*$ is true or not. If true, accept P_i^* else go to Initiate.

E. DIGITAL SIGNATURE GENERATION (DSG)

In this section, we propose our signature generation algorithm based on the elliptic curve cryptosystem. We adopted the dynamic generation of signature which largely depends on the content of the message to be transmitted. Thus, no two signatures are the same making the digital signatures different in every transaction transmitted. DSG algorithm runs in three stages namely ***Setup*****,** ***Generate*** and ***Validate***.The ***Setup*** and ***Generate*** occur on the issuing device whilst ***Validate*** occurs on the device of the intended recipient.

Setup The ***Setup*** phase of the DSG algorithm is initiated on any of the two conditions:

1. When the ***Initiate*** process in PKRG returns valid for a public key P_i.
2. When ***Validate*** returns a new and Validated public key P_i^*.

Given any of the above conditions are met in the presence of the key pair $\{S_i, P_i\}$

1. U_i selects any nonce $\{x, y, z\} \in Z_p$ and forwards to the next phase.

Generate: Given the user identity ID_i, private key S_i, public key P_i, and message m.

1. Compute $I_t = H_3(m), F = (I_t.H_4(ID_i).P) \in G$
2. Compute $D = (F.P_i.z.S_i) \in \mathbb{Z}_p$
3. The generated signature on message m is $Sig_i = (F, D)$

Validate: The recipient runs the Validate upon receiving the transaction, T.

Given identity ID_i, signature Sig_i public key P_i

1. Compute $I_t = H_3(m)$,

$$F = (I_t.H_3(ID_i).P)$$

2. Verify the validity of the signature if $e(Dy, Gz) = e(F.P_i, G)^{yz}$ holds. Accept I_t and Sig_i if the statement holds else reject the transaction, T.

Given P_i, Sig_i and I_t are valid

1. U_i selects random nonce $x, y, z \in Z_p$, retrieves the public key of the intended recipient P_j and computes:

$$T = H_1[x||P_i||E_{Pj}\{ID_i||Sig_i||m||I_t\}||P_j]$$

T together with D are then transmitted to U_j.

2. Once U_j receives the message he decrypts it using his secret key, S_j, retrieves the user identity U_i, verifies the signature Sig_i and updates the old public key with the new one if a newly generated key is found in the transaction different from he has in his distributed ledger.
3. After U_j has accessed the information in the transaction, an acknowledgment has to be sent to the $U_i.U_i$ does not send any other message to U_j relative to this transaction until an acknowledgment has been received.
4. The acknowledgment message takes the same form as the transaction. The content of the message will be to acknowledge the receipt of the message referencing the index I_t of the earlier received transaction from U_i. U_j first calls PKRG to initiate the various phases of the algorithm: ***Initiate*****,** ***Generate*** and ***Validate*** and subsequently calls DSG to generate a signature as well as compute I_t for the acknowledgement through ***Setup*****,** ***Generate*** and ***Validate***.
5. Once the PKRG has either returned P_j^* or maintained P_j, and DSG has Validated Sig_j, U_j selects random nonce $x, y, z \in \mathbb{Z}_p$, and retrieves his earlier computed signature Sig_j and public key of the intended recipient P_i and computes:

$$Ack = H_2[z||P_j||E_{P_i}\{ID_j||Sig_j||\mathrm{m}||I_t\}||P_i]$$

Elliptic Curve Cryptosystem supports reverse mode operation. The private key of any user can be used to Validate any public key generated by the user. When U_j receives a transaction with the message encrypted with a decayed or expired public key P_j, since the secret key S_j is integral in the generation of all the public keys, it can be used to decrypt the message.

TABLE 2. Proposed scheme algorithm.

Algorithm: **Generation of Public Key, Digital Signature Transaction and Acknowledgment**

1. **Elliptic Curve** : $y^3 = x^2 + ax + b$
2. **Public Parameters** $=\{H_1, H_2, H_3, H_4, P, Z_p, G, p\}$
3. $\mathbb{Z}_p \ni \{x, y, z\}$ where x, y, z are nonces
4. Given the secret keys of U_i and U_j are S_i and S_j
5. Given the identities of U_i and U_j are ID_i and ID_j

Generation of Public key P_i

6. Start
7. *If* $(\lambda = 0)\{$
8. *Randomly select* $x, y, z \in Z_p$
9. *Compute*
10. $P_i^* = (S_i)^x \ G \bmod p$
11. *If* $(P_i = P_i^*)\{$
12. *Go to Start}* else{
13. *Proceed}*;
14. *} else {*
15. *Ignore}*;

Generation of Digital Signature and Transaction

Start

Randomly select $\{x, y, z\} \in Z_p$,

16. *Compute*
17. $I_t = H_3(m)$, $F = (I_t . H_4(ID_i) . P)$, $D = (f . P_i . z . S_i)$
18. $Sig_{U_i} = (F, D)$
19. if $e(Dy, Gz) = e(F . P_i, G)^{yz}$ {
20. *Accept* I_t *and* $Sig_{U_i}\}$
21. *Compute* $T = H_1[\mathrm{x} \| P_i \| E_{Pj}\{ID_i \| Sig_{U_i} \| \mathrm{m} \| I_t\} \| P_j]$
22. *Else{*
23. *Reject T}*

Generation of Acknowledgment

24. *Given* U_j *confirms the validity of* Sig_{U_i}, ID_i, P_i
25. U_j *computes* P_j *as in Steps 6-15*
26. U_j *computes* Sig_{U_j} *as in Steps 16-20*
27. *Given* I_t *and* Sig_{U_j} *are valid*
28. *Compute* $Ack = H_2[\mathrm{c} \| P_j \| E_{P_i}\{ID_j \| Sig_{U_j} \| \mathrm{m} \| I_t\} \| P_i]$

IV. CRYPTANALYSIS OF PRIVACY ENHANCEMENT SCHEME

Our scheme was designed to maximize the protection of the credentials of users while ensuring there are little or no traces and patterns for any adversary to connect leading to the identity of the user being revealed. We provide cryptanalysis of our scheme to prove that our scheme is robust enough to provide the services as desired.

A. THREAT MODEL

The viability of our scheme can only be accessed based on a threat model consistent with reality [31]. In our threat model we consider adversary $\mathcal{A}$ as an attacker or intruder that initiates attacks on our network in order to impersonate an honest user as seen in any edge computing environment [1]. Usually, attackers are between the senders and the intended recipients observing traffic and intermittently initiating attacks over the traffic on either a secure or insecure network [3]. However, to demonstrate the efficiency and robustness of our scheme to withstand any form of attack, a polynomial time adversary $\mathcal{A}$ controls all messages transmitted over the network and can therefore delete, tamper, re-route or replay captured messages and has knowledge of parameters declared public on the network. It is assumed the channel between two nodes is

TABLE 3. Summary of proposed scheme.

User U_i	Private Communication	Mobile Device
Step 1 U_i enters $\{S_i, PW_i, f_i\}$ into the mobile device to initiate the computation of transaction.	$\{S_i, PW_i, f_i\}$ →	**Step 2** Verifies the validity of $\{S_i, PW_i, f_i\}$. If all three input parameters are correct, the validity of P_i is then checked else login request is rejected. If $\lambda = 0$, compute P_i^*, else use P_i. Assuming $\lambda = 0$, select $\{x, y, z\} \in \mathbb{Z}_p$ and compute, $P_i^* = (S_i)^x G \ mod \ p$
	← P_i^*	

User U_i	**Insecure Channel**	**User U_j**
Step 2 **U_i selects $\{x, y, z\} \in \mathbb{Z}_p$ in the presence of S_i, P_i and computes the following:** $I_t = H_3(m)$ $F = (I_t . H_4(ID_i) . P) \in G$ $D = (F . P_i . z . S_i) \in \mathbb{Z}_p$ $Sig_i = \{F, D\}$ $T = H_1[x \parallel P_i \parallel E_{PJ}\{ID_i \parallel Sig_{U_i} \parallel m \parallel I_t\} \parallel P_j]$	$\{T, D\}$ →	**Step 4** Decrypt T with S_j and compute the following: $I_t = H_3(m)$ $F = (I_t . H_4(ID_i) . P)$ Verifies if $e(Dy, Gz) = e(F . P_i . G)^{yz}$ is true. If true accept T and compute Ack, else reject. U_j calls PKRG and DSG to compute P_j and S_{U_j}. U_j selects $\{x, y, z\} \in \mathbb{Z}_p$ and computes
	← $\{Ack, D\}$	$Ack = H_2[z \parallel P_j \parallel E_{P_i}\{ID_j \parallel Sig_{U_j} \parallel m \parallel I_t\} \parallel P_i]$
Step 5 **U_i decrypts Ack with S_i and confirms the validity of S_{U_j}. If S_{U_j} is valid, accept Ack and continue transaction with U_j else reject Ack and cease communication with U_i.**		

unsecure and unreliable. All encrypted messages labeled secret cannot be readily decrypted by $\mathcal{A}$ in polynomial time but is allowed to capture any information leakage. Private keys and nonces have high entropy and cannot be guessed by adversary $\mathcal{A}$ in polynomial time. Once a node is compromised, and $\mathcal{A}$ manages to access the secret key and password of the user, he cannot initiate any attack or transaction in the absence of the user's biometric input.

B. USER ANONYMITY AND UNTRACEABILITY

In our scheme, we minimized the use of the chosen identity ID_i of user U_i in the network. Until U_j can decrypt

$$T = H_1[a||P_i||E_{Pj}\{ID_i||Sig_{U_i}||\mathrm{m}||I_t\}||P_j], U_j$$

will not have access to ID_i and Sig_{U_i}. The random generation of the P_i as well as the use of random nonces $x, y, z \in \mathbb{Z}_p$ in the exchange of each transaction, ensured that every transmitted transaction between two users is fresh. Thus, no two sessions between two entities are the same. This makes it difficult for adversary $\mathcal{A}$ to connect patterns and traces to reveal the true identity of any user. Our proposed scheme provides user untraceability while ensuring the anonymity of users as well.

C. INTEGRITY OF TRANSACTION

As stated earlier, encryption alone does not guarantee that captured messages will not be tampered with. Encrypted messages can be captured, re-routed or even replayed at a convenient time to the adversary. To deal with this limitation, we indexed every transaction with I_t. The previous hashed indexes were referenced in every subsequent transaction to enable parties involved to realize whenever a transaction has been captured, replayed, etc. The inclusion of the signature provided another level of integrity to the transaction exchanged between the two parties. Therefore our scheme ensured that the integrity of every transaction is not compromised.

D. MUTUAL AUTHENTICATION

Privacy cannot be achieved without ensuring users are sharing their sensitive information with the right people. Mutual authentication between users was dependent on four factors: identity, signature, public key and index. A malicious user will find it nearly impossible to subvert all the factors. Our scheme provides two levels of authentication for every user to Validate the sender of a transaction. A sender of any message has to encrypt it with the public key of the intended recipient as in

$$E_{P_j}\{ID_i||Sig_{U_i}||Data||I_t\}. \quad (1)$$

The recipient rejects the transaction if he is unable to decrypt the transaction with his private key. Since it is easy for any internal adversary to access any public key, the second level of authentication has to deal with the validation of the sender's identity ID_j and digital signature Sig_j as well as the referencing of the previous hashed transaction index I_t. If any of these elements is invalid, the transaction is rejected. Our scheme provides a comprehensive authentication to protect users against both internal and external attackers.

E. USER IMPERSONATION ATTACKS

Our scheme makes it difficult for $\mathcal{A}$ to impersonate any of the users. S_i and PW_i of U_i are securely stored in the memory of the device and cannot be retrieved even in the case of theft. Furthermore, ***Initate*** takes as input (S_i, PW_i) to either maintain the existing public key P_i if it is still valid or transition into the ***Generate*** if P_i is invalid. Hence, A can not initiate the PKRG without the U_i' s credentials.

F. INSIDER ATTACKS

One major aim of our scheme was to make it difficult for an authorized user to be able to connect transaction patterns leading to the revelation of the identity of any user. No user on the network will be able to know the identity connected to a public key since the identity is encrypted in the message, $E_{Pj}\{ID_i||Sig_{U_i}||\mathrm{m}||I_t\}$. Only U_j will be able to know the identity linked with a generated public key once he successfully decrypts the message. Thus, honest but curious insiders will see different random public keys for an unknown identity, therefore, making it difficult for a successful trace.

V. SECURITY ANALYSIS USING AVISPA SPAN

The robustness of our scheme against any probable attack was simulated using the Automated Validation of Internet Security Protocols and Applications (AVISPA). The AVISPA platform provides applications for verifying and analyzing security protocol to ascertain whether the security protocol is safe for deployment in the real world. AVISPA supports the programming language High Level Protocol Specification Language (HLPSL). AVISPA translates protocols written in HLPSL to Intermediate Format (IF) using a hlpsl2if translator to make the protocol accessible to the backends of the applications [32]. IF through the backends then generate an output format (OF) comprising of the various parts [33]:

1. *SUMMARY*: This section of the OF indicates whether the scheme tested is secure, insecure or indecisive.
2. *DETAILS*: This declares the protocol to safe, susceptible to attacks or outcome indecisive.
3. *PROTOCOL*: Specifies the protocol name.
4. *GOAL*: This section specifies the goal of the simulation.
5. *BACKEND*: The selected backend among *the four used in the test is stated.*
6. *COMMENTS & STATISTICS*: This section displays if there is any attack trace in the scheme.

On-the-fly Model-Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and Tree Automata-based in Automatic Approximation for the Analysis of Security Protocols (TA4SP) are the four backends of the AVISPA. A detailed descrip-

```
role user_1 (Ui, Uj: agent, Pi, Pj: public_key, Snd, Rcv: channel (dy))
played_by Ui def=

local State : nat,
P, Na, Nb, Nc, N, G, It, D, F: text,
IDi, IDj, T, Ack, Sigi, M, Si: message,
H, H1, H2, H3, Mod: hash_func

init
State :=0
transition

1. State = 0 ⋀ Rcv(start) =|> State':= 2
⋀ Na' := new()
⋀Pi' :=(Si.G).Na.Mod(N)
⋀It' := H1(M)
⋀F' :=(It'.H3(IDi).P)
⋀D' :=(F'.Pi'.Na.Si)
⋀Sigi' :=(F'.D')
⋀T' :=H1(Na'.Pi'.{(IDi.Sigi'.M.It')_Pj}.Pj)
⋀Snd(T')
⋀secret(Si,sec1,Ui)
⋀ secret({Sigi', IDi},sec2,{Ui,Uj})
⋀ witness(Ui,Uj,uj_ui_t,T')

2. State = 2 ⋀ Rcv(Ack') =|> State':= 4
⋀ request(Ui,Uj,ui_uj_ack,Ack')
end role
```

FIGURE 2. Role specification for *Ui* in HLPSL.

```
role user_2(Ui, Uj: agent, Pj, Pi: public_key, Snd, Rcv: channel (dy))
played_by Uj def=

local State : nat,
P, Na, Nb, Nc, N, G, It, D, F: text,
IDi, IDj, T, Ack, Sigj, M, Sj : message,
H1, H2, H3, Mod: hash_func

init
State := 1
transition

1. State = 1 ⋀ Rcv(T') =|> State':= 3 ⋀ Nb' := new()
⋀Pj' :=(Sj.G).Nc.Mod(N)
⋀It' :=H1(M)
⋀F' :=(It'.H3(IDj).P)
⋀D' :=(F'.Pj'.Nb'.Sj)
⋀Sigj' :=(F'.D')
⋀Ack' :=H3(Nb'.Pj'.{(IDj.Sigj'.M.It')_Pi}.Pi)
⋀Snd(Ack')
⋀secret(Sj, sec3, Uj)
⋀ secret({Sigj', IDj},sec4,{Ui,Uj})
⋀ witness(Uj,Ui,ui_uj_ack,Ack')
end role
```

FIGURE 3. Role Specification for U_j in HLPSL.

```
role session(Ui, Uj: agent, Pi, Pj: public_key) def=
local SA, RA, SB, RB: channel (dy)
composition
user_1(Ui,Uj,Pi,Pj,SA,RA)
⋀user_2 (Ui,Uj,Pi,Pj,SB,RB)
end role
```

FIGURE 4. Role specification for *Session* in HLPSL.

tion of four backends of AVISPA is extensively discussed in [34], [35].

We modeled our simulation using Dolev – Yao (*dy*) channel with the intruder denoted *i* to test the vulnerability of the proposed scheme relative to the activities of the intruder. Under this model, *i* has full control over the network with full access to messages exchanged between U_i and U_j. Intruder *i* can intercept, analyze, modify, compose any message and subsequently forward to any user as far as he knows the required keys [32]. The aims of the simulation:

1. Dolev-Yao model check
2. Replay attack check
3. Verify the executability check in non-trivial HLPSL specification

A. DESCRIPTION OF THE HLPSL IMPLEMENTATION

As stated, two basic roles were involved in the verification of the scheme's robustness against attacks and manipulation. These are *user*_1 played by *Ui* and *user*_2 played by *Uj*. In addition to the basic roles, we declared three other roles as required in HLPSL namely *session*, *goal* and *environment*. The basic user roles were defined as seen in Fig. 2 and Fig. 3. As shown in Fig. 2, *Ui* generated a new random number *Na* and computed the values of a new public key *Pi*, index *It, F*, *D* and the digital signature *Sigi*.

With all these parameters successfully computed, *Ui* then composed the transaction, T and forwarded it to U_j using the *Snd* command. In the declaration of role*user*_1, we specified the intruder *i* should not have access to the identity *IDi* and signature *Sigi* through the expression $secret(\{Sigi', IDi\}, sec1, \{Ui, Uj\})$ since these cryptographic primitives are encrypted with *Pj*. Our scheme is flawed if this requirement is breached. *Uj* authenticated the validity of *Ui* once he received T based on the expression $witness(Ui, Uj, uj\_ui\_t, T')$.

Likewise, *Ui* authenticated the validity of *Uj* upon receiving *Ack* through the expression $(request(Ui, ui\_uj\_ack, Ack')$.

The declaration of *user*_2 played by *Uj* is done in a similar fashion as seen in Fig. 2. However, *Uj* transitioned from his initial *State* 1 *to 3* when he received T transmitted by U_i.

The role environment instantiated all the global constants. The intruder *i* is declared as an additional legitimate user with parameters assigned to it as the knowledge *i* has. In this case, all hash functions, public keys, and his private key k_i formed the knowledge of the intruder. We declared the goals of the simulation within this role. The proposed scheme had two authentication goals and four secrecy goals as seen in figure 6. *Secrecy_of sec1* indicated that only U_i should have access to S_i. *Authentication_onuj_ui_ack* means that U_j has computed *Ack* for U_i. U_i then checked the validity of U_j based on *Ack*.

B. DISCUSSION OF SIMULATION RESULTS

Our proposed scheme was simulated under the OFMC and CL-AtSe backends of AVISPA. The discussions of the results are done vis-à-vis the goals of the simulation as earlier listed.

- Dolev-Yao model check: The backends check the possibility of the intruder *i* to initiate a man-in-the-middle attack. The results from both backends indicate that the proposed scheme is robust against this attack and is therefore safe.
- Replay attack check: The intruder *i* is offered details of sessions between U_i and U_j while these parties exchange transactions. Results shown in figures 6 and 7 indicate that, despite all the information *i* has, our scheme is once again robust when it comes to replay attacks.

```
role environment() def=
const ui, uj      : agent,
pi, pj, ki  : public_key,
h1, h2, h3: hash_func,
sec1, sec2, sec3, sec4,
ui_uj_ack,
uj_ui_t : protocol_id
intruder_knowledge = {h1, h2, h3,ui, uj,  pi, pj, ki, inv(ki)}
composition
session(ui,uj,pj,pi)
/\session(ui,i,pi,ki)
end role

goal
secrecy_of sec1
secrecy_of sec2
secrecy_of sec3
secrecy_of sec4
authentication_on ui_uj_ack
authentication_on uj_ui_t

end goal

environment()
```

FIGURE 5. Role specification for environment in HLPSL.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/PES.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.08s
 visitedNodes: 10 nodes
 depth: 3 plies
```

FIGURE 6. Results of test using OFMC backend.

- Executability check: This test checks to see if all the declared goals as stated in the simulation in *role environment* were achieved. Once the output from both backends did not indicate *i* had access to any of the parameter declared *secret* thereby making our scheme unsafe, then our scheme passed this test.

VI. PRACTICAL SIMULATION AND DISCUSSION OF SIMULATION RESULTS

In this section, we discuss the practical simulation of our scheme in an NS2 2.35 environment. NS2 is an open-source widely accepted network simulator that can be used to simulate the practicality of network types, protocols, and various traffic models. In our simulations, the parameters used are listed in the table

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/PES.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed  : 1 states
 Reachable  : 1 states
 Translation: 0.02 seconds
 Computation: 0.00 seconds
```

FIGURE 7. Results of test using CL-AtSe backend.

TABLE 4. Simulation parameters.

Parameter	Description
OS	Kubuntu 18.04.2 LTS
Numbers of users used	10 (Scenario 1) 20 (Scenario 2)
Simulation time T_s	1800 seconds
Initial energy e_i of each user	100J
Mobility of users	2m (Scenario 1) 5m (Scenario 2)
Public Key expiration time λ	60 seconds (Scenario 1) 180 seconds (Scenario 2)

We limited the simulation of our scheme to the transmission of a transaction by one user as well as the sending of the acknowledgment of receipt by the recipient. All the length of the hash outputs were assumed to be 160 bits, random nonces were assumed to be 128 bits while the I_t was assumed to be 32 bits. Users were required to broadcast their initial public keys to all nodes in the network. After that, the randomly generated public keys were exchanged between only two users at a time. In each scenario, we have two messages been transmitted $M_1 = T$ from U_i to U_j and $M_2 = Ack$ from U_j to U_i. Thus both transmitted messages M_1 and M_2 were assumed to be 480 bits long.

The simulation focused on three key parameters throughput (bps), load (bps) and energy consumption (mW). These were the main parameters analyzed considering the resource

limitations of our nodes. The analysis was done based on the calculation involving Total Packets Sent (S_p), Total Packets Received (R_p), Bit Size of Packet (P_s), Simulation Time (T_s), initial energy (E_i) and remaining energy (E_r).

A. LOAD ANALYSIS

We calculated the average load on the nodes as they executed the scheme. The load was calculated as

$$Load(bps) = \frac{(S_p + R_p) \times P_s}{T_s}$$

The values of the load are 42.11 bps and 53.09 bps for scenarios 1 and 2 respectively. It was observed that in both scenarios, the load was relatively high when nodes initially broadcasted packets (public keys) to all nodes. However, load on the nodes was low while transactions were exchanged between just two users. An increase in the nodes in scenario 2 saw a slight increase in the average load in the network largely due to the earlier broadcast packets.

B. THROUGHPUT ANALYSIS

Throughput can be defined as the number of transmitted bit per unit time. Throughput was calculated as

$$Throughput(bps) = \frac{R_p \times P_s}{T_s}$$

The average throughput values in the simulation are 49.91 bps and 62.36 bps for scenarios 1 and 2 respectively. Similarly, the throughput increases with an increase in the number of nodes since the exchanged transactions increase with an increase in number nodes.

C. ENERGY CONSUMPTION ANALYSIS

It was important to analyze how much energy nodes consume while running our proposed scheme. This parameter was calculated using the formula

$$Energy\ Consumption\ (mW) = \frac{E_i - E_r}{T_s}$$

The average energy consumed by all the nodes is 9.43 mW and 16.29 mW respectively for scenarios 1 and 2. The summary of all the simulation results is displayed in figure 8.

VII. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of our scheme and compare the evaluation to two similar protocols proposed by Reddy et al [33] and Shivanna et al [36]. Alavapati et al. proposed an enhanced two-factor authentication scheme based on ECC. To achieve privacy preservation, Shivanna et al. also proposed privacy enhancement scheme based on double encryption of user data. The evaluation was done primarily based on the security parameters, computational cost, and communication overhead. Despite the difference in deployed environments and the number of agents involved in the protocol execution, the aim of both protocols is similar with much focus on mutual authentication, user anonymity, and untraceability. Nonetheless, our proposed

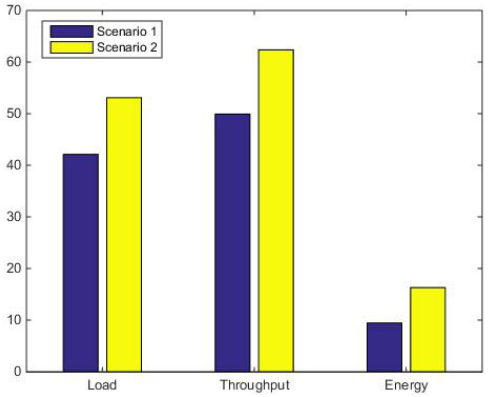

FIGURE 8. Simulation results of PES.

TABLE 5. Comparison of security features.

Security Feature	Alavalapati et al. [33]	Shivanna et al. [36]	Proposed Scheme
R_1	Yes	Moderate	Yes
R_2	Yes	No	Yes
R_3	Yes	Yes	Yes
R_4	Yes	No	Yes
R_5	Moderate	No	Strong
R_6	Moderate	No	Strong
R_7	Yes	No	Yes
R_8	No	Yes	Yes
R_9	Moderate	Yes	Yes

scheme has additional security features that tend to make ours more comprehensive and robust. Our proposed scheme, however, went a step further to put mechanisms in place to ensure any form of manipulation of transmitted transactions is easily detected.

R_1 : Provision of user anonymity/ untraceability
R_2 : Resistance to impersonation attacks
R_3 : Provision of mutual authentication
R_4 : Prevent insider attack
R_5 : Prevent replay attack
R_6 : Prevent man-in-the-middle attack
R_7 : Prevent user impersonation attack
R_8 : Provides user credential privacy
R_9 : Provision of data security

Both schemes proposed by Alavalapati et al. and Shivanna et al. involved three agents compare to the two in our scheme. In Table 5, we provide a summary of the computational cost involved in the execution of the three protocols among their agents. The comparison was done based on the method used in [33].

T_h : Time complexity of a one-way function
T_{mul} : Time complexity of a point multiplication operation on elliptic curve
T_{fun} : Time complexity of encryption or decryption function
T_{exp} : Time complexity of an exponential function

Since we have different numbers of agents and our scheme provided some additional security features, the computation

TABLE 6. Computation and overhead cost comparison.

Agent	Alavalapati et al. [33]	Shivanna et al. [36]	Proposed Scheme
$MU/U/U_i$	$10T_h + 3T_{mul}$	$11T_h + 4T_{mul}$	$3T_h + 2T_{mul} + 2T_{fun} + 2T_{exp}$
$FA/CS/U_j$	$5T_h + 2T_{mul}$	$5T_h + 2T_{mul}$	$3T_h + 2T_{mul} + 2T_{fun} + 2T_{exp}$
HA/DC	$7T_h + 2T_{mul}$	$8T_h + 2T_{mul}$	N/A
Total	$22T_h + 7T_{mul}$	$24T_h + 8T_{mul}$	$6T_h + 4T_{mul} + 4T_{fun} + 4T_{exp}$

cost was different as shown in the table. However, our scheme showed its efficiency on cost on common metrics such as T_h and T_{mul}. The provision of encryption and decryption within our scheme and the computation of the random public keys accounted for the other metrics in our scheme. Overall, our scheme requires lesser computations in providing more security services compared to the scheme proposed in [33].

VIII. CONCLUSION

Privacy issues in edge computing has come to the fore due to the involvement of third parties service delivery agents in the environment. Despite the improvements anticipated to be witnessed due to the integration of blockchain principles in the various paradigms of edge computing, the principle of transparency in blockchain also presents issues relative to user untraceability and anonymity. In our scheme, we discussed the limitation of using only encryption as a means of enhancing privacy. We proceeded to introduce our scheme which is based on the random generation public keys and digital signature generation based on the content of the message to be transmitted. We demonstrated that our scheme is efficient to achieve user anonymity and untraceability while not compromising the integrity of the transactions traversing the network. We tested our scheme on AVISPA, with the results clearly showing that our scheme is safe and robust to withstand any form of replay and man-in-middle attacks. Practical simulations undertaken in NS2 also showed our scheme is efficient and low on resources. A performance analysis was done compared with a scheme proposed by Alavalapati *et al.* [33] and Shivanna *et al.* [36]. The comparative analysis shows our scheme has a very low computational cost than the proposed schemes in [33], [36].

REFERENCES

[1] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[2] M. Maroufi, R. Abdolee, and B. M. Tazekand, "On the convergence of blockchain and Internet of Things (IoT) technologies," 2019, *arXiv:1904.01936*. [Online]. Available: https://arxiv.org/abs/1904.01936

[3] K. Bhardwaj, M.-W. Shih, A. Gavrilovska, T. Kim, and C. Song,"SPX: Preserving end-to-end security for edge computing," 2018, *arXiv:1809.09038*. [Online]. Available: https://arxiv.org/abs/1809.09038

[4] D. Wang, H. Cheng, D. He, and P. Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Syst. J.*, vol. 12, no. 1, pp. 916–925, Mar. 2018.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[6] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Montana, MT, USA: Springer, 2015.

[7] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security Privacy*, vol. 8, no. 6, pp. 24–31, Nov./Dec. 2010.

[8] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and solutions," 2016, *arXiv:1608.05187*. [Online]. Available: https://arxiv.org/abs/1608.05187

[9] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[10] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. Conf. Internet Meas. Conf.*, 2013, pp. 127–140.

[11] K. Knibbs. (Jul. 7, 2015). *A Friendly Reminder: Bitcoin Is Not Anonymous*. [Online]. Available: https://gizmodo.com/a-friendly-reminder-bitcoin-is-not-anonymous-1682885318

[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology. CRYPTO 1984* (Lecture Notes in Computer Science), vol 196, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985.

[13] Z. Xia, L. Zhang, and D. Liu, "Attribute-based access control scheme with efficient revocation in cloud computing," *China Commun.*, vol. 13, no. 7, pp. 92–99, Jul. 2016.

[14] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology-EUROCRYPT'98* (Lecture Notes in Computer Science), vol. 1403. Espoo, Finland: Springer, 1998, pp. 127–144.

[15] R. N. Lakshmi, R. Laavanya, M. Meenakshi, and C. S. G. Dhas, "Analysis of attribute based encryption schemes," *Int. J. Comput. Sci. Eng. Commun.*, vol. 3, no. 3, pp. 1076–1081, 2015.

[16] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, and R. Buyya, "Attribute-based data access control in mobile cloud computing: Taxonomy and open issues," *Future Gener. Comput. Syst.*, vol. 72, pp. 273–287, Jul. 2017.

[17] W. W. Zhang, R. Zhang, J. Y. Liu, X. X. Niu, and Y. X. Yang, "Efficient chosen-ciphertext secure proxy re-encryption scheme," *J. Commun.*, vol. 34, no. 7, pp. 87–97, 2013.

[18] Microsoft. (Nov. 8, 2019). *Homomorphic Encryption*. [Online]. Available: https://www.microsoft.com/en-us/research/project/homomorphic-encryption/

[19] C. Gentry, "A fully homomorphic encryption scheme," in *Department of Computer Science*, Ann-Arbor, MI, USA: Stanford Univ., 2009

[20] M. Ogburn, C. Turner, and P. Dahal, "Homomorphic encryption," *Procedia Comput. Sci.*, vol. 20, pp. 502–509, 2014.

[21] H. Byun. (2019). *The Advantages and Disadvantages of Homomorphic Encryption*. Accessed: Sep. 7, 2019. [Online]. Available: https://baffle.io/blog/the-advantages-and-disadvantages-of-homomorphic-encryption/

[22] R. Amin, S. H. Islam, G. Biswas, D. Giri, M. K. Khan, and N. Kumar, "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4650–4666, Nov. 2016.

[23] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, Aug. 2011.

[24] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.

[25] N. Alexopoulos, J. Daubert, M. Mühlhäuser, and S. M. Habib, "Beyond the hype: On using blockchains in trust management for authentication," in *Proc. IEEE TRUSTCOM*, Aug. 2017, pp. 546–553.

[26] R. Gaal. (Jan. 3, 2019). *Blockchain, the Next Authentication Provider?* [Online]. Available: https://www.traxion.com/en/blockchain-the-next-authentication-provider/
[27] A. Moinet, B. Darties, and J. L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," 2017, *arXiv:1706.01730*. [Online]. Available: https://arxiv.org/abs/1706.01730
[28] K.-A. Shim, "An ID-based aggregate signature scheme with constant pairing computations," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1873–1880, Oct. 2010.
[29] *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, American Bankers Association, Washington, DC, USA, 1999.
[30] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[31] 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. Inf.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.
[32] The AVISPA Team. *HLPSL Tutorial—A Beginner's Guide to Modelling and Analysing Internet Security Protocol*. Accessed: Nov. 3, 2006. [Online]. Available: http://www.avispa-project.org/2006
[33] A. G. Reddy, A. K. Das, E.-J. Yoon, and K.-Y. Yoo, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.
[34] AVISPA. *Automated Validation of Internet Security Protocols and Applications*. Accessed: Jul. 2019. [Online]. Available: http://www.avispa-project.org/
[35] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.
[36] K. Shivanna, S. P. Deva, and M. Santoshkumar, "Privacy preservation in cloud computing with double encryption method," *Computer Communication, Networking and Internet Security* (Lecture Notes in Networks and Systems), vol 5, S. Satapathy, V. Bhateja, K.Raju, and B. Janakiramaiah, Eds. Singapore: Springer, 2017.

BONNAH ERNEST received the B.Sc. and M.Sc. degrees from the Kwame Nkrumah University of Science and Technology, Ghana. He is currently pursuing the Ph.D. degree with the School of Computer Science and Telecommunication Engineering, Jiangsu University. His research interests include security in edge computing, wireless sensor networks, and cyber physical systems.

JU SHIGUANG received the M.S. degree from the Nanjing University of Science and Technology, China, in 1988, and the Ph.D. degree from the CINVESTAV, National Polytechnic Institute, Mexico, in 1996. He is currently a Professor with the School of Computer Science and Communication Engineering, Jiangsu University, China. His current research interests include wireless networks, information security, and big data.