Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-cloud Trust Integrity-based Security Management Architecture for Blockchain-Enhanced Secure Network Slicing Using CKBNA and SFDA

Pullaiah Chowdary Vutla

vutlachowdary@gmail.com

University of the Cumberlands

Triveni Yenugu

Jawaharlal Nehru Technological University Hyderabad

Research Article

Keywords: Secure Network Slicing, Multi-Cloud, Future Cyber Warfare, Blockchain, Security Management Architecture (SMA), Post-Quantum Cryptographic (PQC), and Artificial Intelligence (AI)

Posted Date: April 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8815124/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

MULTI-CLOUD TRUST INTEGRITY-BASED SECURITY MANAGEMENT ARCHITECTURE FOR BLOCKCHAIN-ENHANCED SECURE NETWORK SLICING USING CKBNA AND SFDA

First author **Mr. Pullaiah Chowdary Vutla**[1], Second author **Miss. Triveni Yenugu**[2]
[1]*Research Scholar, Department of Information Technology, The University of the Cumberlands, Williamsburg, KY 40769 USA*
[1]Corresponding author Email ID: vutlachowdary@gmail.com
[2]*Computer Science and Engineering (CSE), Jawaharlal Nehru Technological University Hyderabad, Telangana 500085, India*
[2]Email ID: *yenugutriveni944@gmail.com*

Abstract: Secure Network Slicing (SNS) assists in isolating user data within dedicated virtualized resources. Nevertheless, prevailing works overlooked misconfiguration in multi-cloud, resulting in cross-cloud data leakage. Hence, this paper proposes a Crystals-Kyber Bickley-Naylor Algorithm (CKBNA)-based Secure Key Establishment (SKE) and Struve Function-based Dilithium Algorithm (SFDA)-based signature verification. Initially, the tenants register into a multi-cloud. After that, the SKE is done. Moreover, using Inverse-polynomial Linear Unit EvoNorm-Long Short-Term Memory (IpLUEN-LSTM), the network slice is predicted. Next, the slice's resource and Identity (ID) are initialized, followed by signature creation using SFDA. Later, the trust is estimated, followed by Hyperbolic Growth Gazelle Optimization (HGGaO)-based network slice deployment. Subsequently, by using the CKBNA ChaCha20 (CKBNAChaCha20) algorithm, the tenant data is secured. Then, the signature is verified, and the tenant's secured data is stored. For tamper-proofing, all the transactions are logged into the blockchain. Therefore, the SKE was done with a 2.467% Shared Key Error Rate (SKER).

Keywords: *Secure Network Slicing, Multi-Cloud, Future Cyber Warfare, Blockchain, Security Management Architecture (SMA), Post-Quantum Cryptographic (PQC), and Artificial Intelligence (AI).*

1. INTRODUCTION

The adoption of network slicing has been led by the evolution of 5th Generation (5G) and the emerging 6th Generation (6G) infrastructure (Benzaid et al., 2022). This assists numerous Artificial Intelligence (AI) applications like industrial automation, smart healthcare, and so on (Tabatabaei et al., 2025). The heterogeneous network is partitioned into 3 network slices, namely enhanced Mobile BroadBand (eMBB), Ultra-Reliable Low Latency Communication (URLLC), and massive Machine-Type Communication (mMTC) (Zamfirescu et al., 2024) (Etienne et al., 2024). The increase in multi-cloud deployment has resulted in security vulnerabilities (Joshi, 2025), even though network slicing has become a virtual resource (Diallo et al., 2025). Therefore, the tenant data is exploited by the adversaries in future cyber warfare (Zeydan et al., 2023). Moreover, the slicing lacks the ID binding and doesn't consider quantum adversaries (Benlloch-Caballero et al., 2025). Additionally, the network slice security is deteriorated by the

absence of tamper-proof audits and trust assessment (Cui et al., 2025). Further, due to improper resource placement, the scalability of the multi-cloud network is affected (Alnaim, 2024). Hence, an AI-based autonomic and scalable Security Management Architecture (SMA) that combines CKBNA and SFDA for SNS is proposed.

1.1 Research Overview

Research Gap: Misconfiguration in multi-cloud environments allows a tenant's data stored in one cloud to be accessed by another cloud, causing cross-cloud data leakage. The prevailing work didn't bind the slice identities, thus facilitating adversaries in future cyber warfare to exploit data. Also, multi-cloud ID integrity is compromised by the emerging quantum attacks.

Problem Statement: The prevailing works' limitations are,

- Prevailing (Wang et al., 2024) lacked tamper-proof audit trails, leading to forged updates and malicious rollbacks.
- In prevailing works, the resource demand was inaccurately predicted, causing inefficient slice orchestration and congestion across slices.
- In the conventional (Zeydan et al., 2022), no trust evaluation existed, resulting in unreliable slice placement, mitigated scalability, and cross-cloud data breach.

Significance and Scope of the Study: Here, a post-quantum secure, blockchain-based, and AI-enabled SMA for multi-cloud SNS is developed. Here, the slice ID is bound, and trust is computed to ensure resilience against adversaries.

Research Aim and Objectives: This research aims to launch a secure, scalable, and robust multi-cloud SNS architecture. The key contributions are,

- CKBNA-based SKE and SFDA-based signature creation and verification are performed to ensure multi-cloud trust integrity and avoid cross-cloud data leakage.
- The IpLUEN-LSTM classifier is utilized to enhance the Network Slice Prediction (NSP) and improve slice orchestration.
- To ensure optimal and trustworthy slice deployment, trust is estimated, and the HGGaO-based slice deployment is done.
- The slice operation and transactions are stored in the blockchain to provide tamper-proof auditability.

The remaining paper is framed as: The related works are illustrated in Section 2, followed by the proposed methodology in Section 3 and performance analysis of the proposed models in Section 4. Then, Section 5 concludes the paper with future scope.

2. LITERATURE SURVEY

(Wang et al., 2024) established privacy protection for network slicing. Initially, for optimizing encryption, the Chameleon Hash Algorithm (CHA) was utilized. Afterthat, the network privacy was effectively protected by the homomorphic encryption. On the other hand, policy manipulation was caused by the absence of audit trails.

(Zeydan et al., 2022) developed post-quantum blockchain-centric network orchestration. Here, the N-th degree Truncated polynomial Ring Units (NTRU) was utilized to secure the tenant data in the multi-cloud network. And, to maintain transparency, the network logs were saved in a blockchain. Yet, the scalable slicing wasn't taken into account.

(Irshad et al., 2023) employed PQC and blockchain-based secure and scalable cloud architecture. Here, using the Multicast and Broadcast Rekeying Algorithm (MBRA), the privacy of the user's information was ensured. The MBRA combined PQC and blockchain. Therefore, the security was ensured. But, the congestion along the slices existed.

(Abdellatif et al., 2023) integrated AI-assisted network slicing for 5G-and-beyond networks. Primarily, for inter-slice allocation, the Proximal Policy Optimization (PPO) was utilized. Next, the Reinforcement Learning-based Inter-Slice Allocation (RLISA) was performed precisely to update the reserved paths. Nevertheless, the cross-cloud data leakage was prevalent.

(Bharot et al., 2025) presented secure data sharing in a multi-cloud environment. Here, the ChaCha20-Poly1305 encryption model was utilized to secure the data. Moreover, to securely share the key betwixt the data owner and authentic users, the Elliptic Curve Diffie-Hellman (ECDH) model was utilized. In contrast, the quantum attacks couldn't be resisted.

3. PROPOSED METHODOLOGY FOR SECURE NETWORK SLICING IN MULTI-CLOUD NETWORK

In Figure 1, the CKBNA-based SKE and SFDA-based signature generation/creation and verification for SNS in a multi-cloud network framework are depicted.

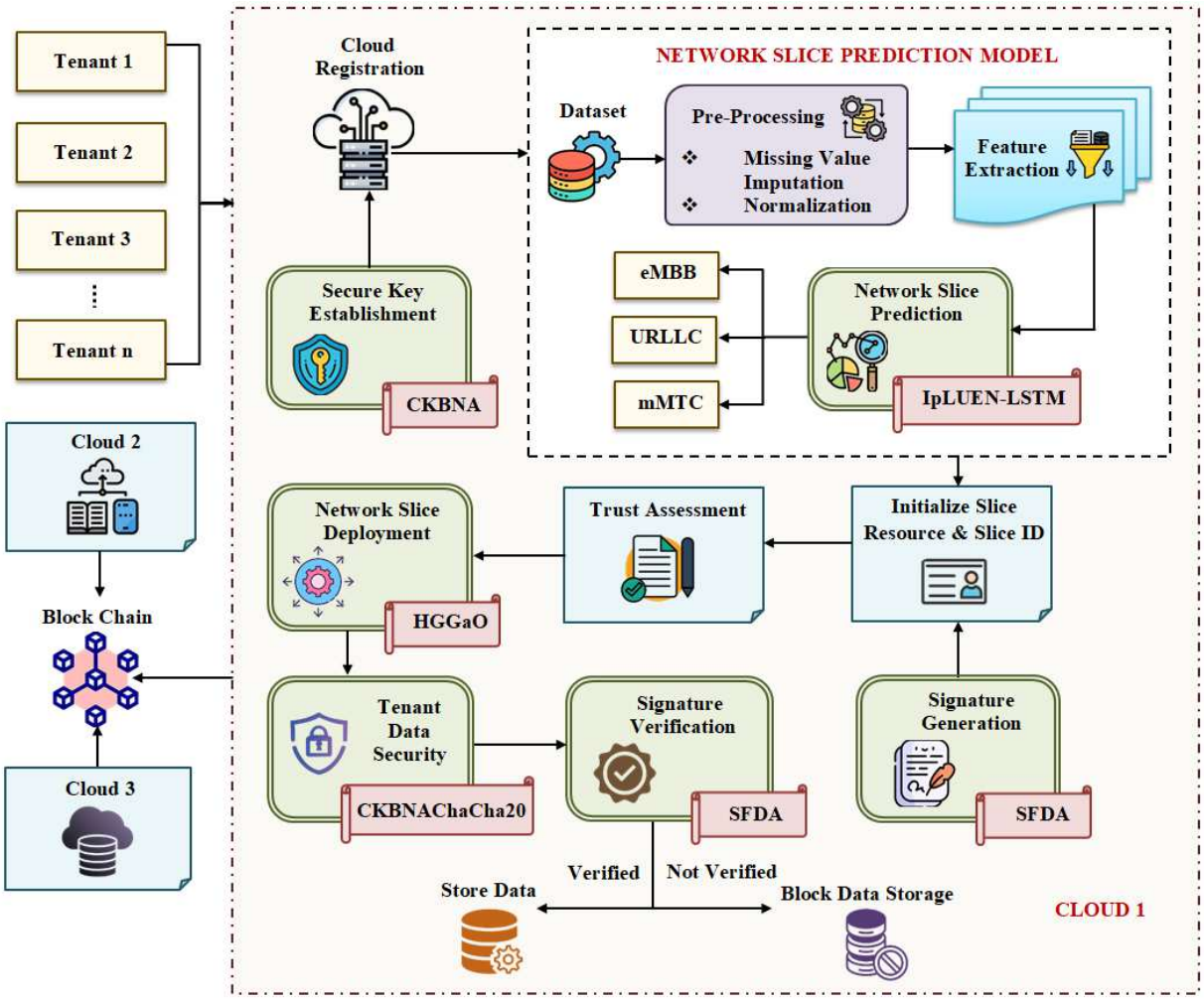


Figure 1: Architecture of the Proposed System

3.1 Cloud Registration & Secure Key Establishment

The proposed framework begins with the registration of tenants into the multi-cloud. Here, by using the tenant ID and password, the tenants (T) register in the cloud.

$$T = \left[T^1, T^2, T^3, ..., T^n\right] \tag{1}$$

Where, the (n^{th})tenant is notated as (T^n). Further, using CKBNA, the SKE is carried out. Here, Crystals-Kyber Algorithm (CKA) is a post-quantum key encapsulation mechanism that is used for SKE. A Centred Binomial Distribution (CBD) is utilized by CKA for generating its noise. Nevertheless, CBD is predictable. Therefore, for generating a non-uniform distribution, the Bickley–Naylor (BN) function is used. Initially, the cloud (C)creates a random polynomial (g). After that, the distribution $(\alpha)$is equated using the BN function, which exhibits non-linear decay.

$$\alpha = \int_{0}^{\pi/2} \exp^{-g/\cos\vartheta} \partial\vartheta \tag{2}$$

Here, $(\pi, \exp, \vartheta)$signifies an irrational number (22/7), an exponential factor, and an angle parameter, respectively. Now, the public key $\left(h^{\text{cloud}}\right)$ at the cloud $(C)$is derived as,

$$h^{\text{cloud}} = [\alpha \times g] + g \tag{3}$$

Now, the tenant (T)produces a shared key $\left(h^{\text{shared}}\right)$ and shares it with (C). This $\left(h^{\text{shared}}\right)$serves as a symmetric key for tenant data security.

$$h^{\text{shared}} = \beta \times \left(h^{\text{cloud}} \| \alpha\right) \tag{4}$$

Where, the key derivative function is notated as (β). Moreover, the network slice is predicted.

3.2 Network Slice Prediction Model

Here, the Network Slice Prediction Model (NSPM) is trained. The step-by-step process is elucidated further.

3.2.1 Dataset

Initially, the 5G network slicing dataset is gathered. The network slice device parameters (P) in the gathered dataset are given as,

$$P = \{P_1, P_2, P_3, ..., P_j\} \tag{5}$$

Where, the $\left(j^{th}\right)$collected data is symbolized as $\left(P_j\right)$.

3.2.2 Pre-Processing

Now, to convert (P)into a structured form, the pre-processing is performed. Primarily, by using Missing Value Imputation (MVI), the missing values in $(P)$are assigned. Here, for imputation, the mean of the neighbouring data is used. Further, using min-max normalization, the imputed data $(P'')$is normalized in the range of 0 to 1.

$$\widetilde{P} = [P'' - P''_{\min}] / [P''_{\max} - P''_{\min}] \tag{6}$$

Where, $\left(\widetilde{P}, P''_{\min}, P''_{\max}\right)$are the normalized data, minimum (P''), and maximum (P''), correspondingly.

3.2.3 Feature Extraction

Further, from $(\widetilde{P})$, the features (A) like data rate, connection type, latency, bandwidth, reliability, mobility level, and so on are extracted. This (A) is fed into IpLUEN-LSTM for NSP.

3.2.4 Network Slice Prediction

Now, using the IpLUEN-LSTM, the network is sliced. Here, the LSTM is chosen for NSP. LSTM models the temporal dynamics and nonlinear behaviors. Nevertheless, saturation in LSTM is caused by the sigmoid activation function. In addition, the training could be unstable in LSTM. Therefore, the Inverse polynomial Linear Unit (IpLU) and the EvoNorm (EN) are used in LSTM. Figure 2 depicts the proposed IpLUEN-LSTM's architecture.

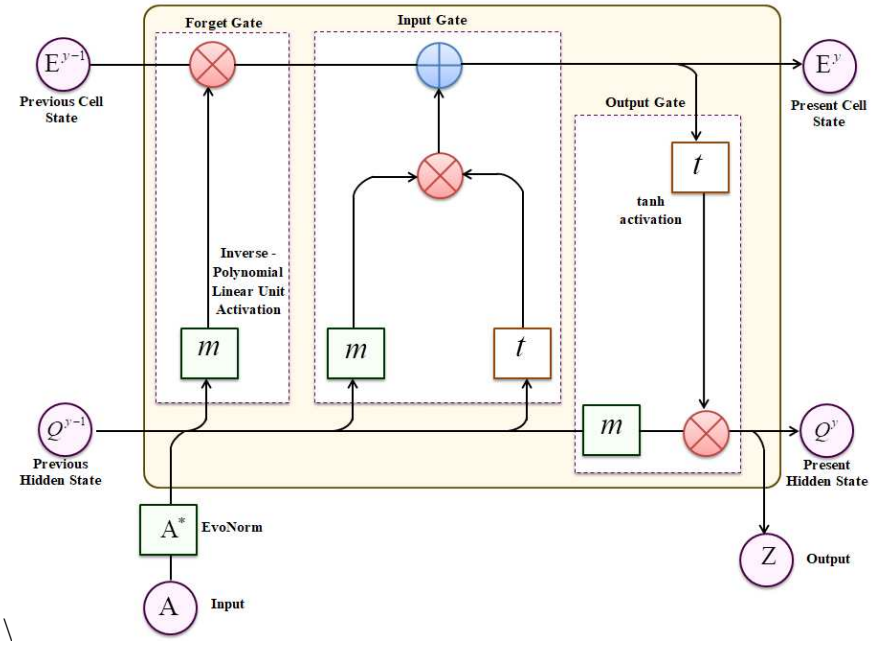


Figure 2: Proposed IpLUEN-LSTM Classifier

Primarily, to make the training more stable, the EN that balances the data across different samples is applied to the input (A).

$$A^{*} = \frac{A}{\sqrt{\sigma^{2}(A) + \delta}} \tag{7}$$

Where, the EN output is signified as (A^{*}) and the variance of the feature vector and the constant for numeric stability are symbolized as (σ, δ), correspondingly. Then, in the forget gate, the data to be added or removed is determined. Hence, the forget gate output (K) is provided as,

$$K = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes E^{y-1} \tag{8}$$

$$m = 1 / \left[1 + \left(\varepsilon \left| A^{*} \right|^{d} \right) \right] \tag{9}$$

Here, the previous hidden state and previous cell state for time (y) are signified as $\left(Q^{y-1}, E^{y-1}\right)$, the weight value, bias value, scaling factor, and the polynomial degree are symbolized as (w, b, ε, d), correspondingly, and the IpLU activation function that learn the long-term dependencies and minimizes the vanishing gradients' risk is denoted as (m). Then, the new data to be added to the present cell state $\left(E^{y}\right)$ is determined through the input gate.

$$S = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times t \right\rangle \tag{10}$$

$$t = \left(\exp^{A^{*}} - \exp^{-A^{*}} \right) / \left(\exp^{A^{*}} + \exp^{-A^{*}} \right) \tag{11}$$

Where, the output of the input gate and the hyperbolic tangent (tanh) activation function are symbolized as (S, t). Now, $\left(E^{y}\right)$is assessed as,

$$E^{y} = K \oplus S \tag{12}$$

Lastly, with respect to $\left(E^{y}\right)$, $\left(A^{*}\right)$, and $\left(Q^{y-1}\right)$, the NSP output (Z) is obtained.

$$Z = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes \left[t \times E^{y} \right] \quad \text{for } \left(\mathrm{Z} = \mathrm{Z}^{1}, Z^{2}, Z^{3} \right) \tag{13}$$

Where, eMBB, URLLC, and mMTC network slices are signified as $\left(\mathrm{Z}^{1}, Z^{2}, Z^{3}\right)$, correspondingly.

Pseudo-code for IpLUEN-LSTM

Input: Extracted Features (A)

Output: NSP Output (Z)

Begin

Initialize $\left(Q^{y-1}\right), \left(E^{y-1}\right), (w, b)$

While $(time \le y)$

For (A)

Balance data $A^{*} = \dfrac{A}{\sqrt{\sigma^{2}(A) + \delta}}$

Determine activation functions

$$m = 1 / \left[1 + \left(\varepsilon \left| A^{*} \right|^{d} \right) \right], t = \left(\exp^{A^{*}} - \exp^{-A^{*}} \right) / \left(\exp^{A^{*}} + \exp^{-A^{*}} \right)$$

Estimate forget gate

$$K = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes E^{y-1}$$

Find input gate

$$S = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times t \right\rangle$$

Update present cell state $E^{y} = K \oplus S$

Exectute output gate

$$Z = \left\langle \left\{ \left[\left(Q^{y-1}, A^{*} \right) \times w \right] + b \right\} \times m \right\rangle \otimes \left[t \times E^{y} \right] \quad \text{for } \left(\mathrm{Z} = \mathrm{Z}^{1}, Z^{2}, Z^{3} \right)$$

End for

End while

Obtain (Z)

End

Hence, the NSPM is trained. The cloud is split into classified slices in a real-time application, and this guarantees that each tenant gets the right slice at the right time.

3.3 Initialize Slice Resource and Slice ID

Following slicing, the slice resources like bandwidth (γ) and the latency budget (η) for $\left(\mathrm{Z}^{1}, Z^{2}, Z^{3} \right)$ are initialized. Further, the slice ID (q) for $\left(\mathrm{Z}^{1}, Z^{2}, Z^{3} \right)$ like $\left(q^{1}, q^{2}, q^{3} \right)$ is initialized. Then, to configure the slice and the cloud, a signature is produced.

3.3.1 Signature Generation

Now, by utilizing SFDA, the signature is produced centered on the slice ID (q) and the cloud ID (i). Here, for signature generation, the Dilithium Algorithm (DA), which is a lattice-centric digital signature scheme for post-quantum security, is selected. Nevertheless, DA gives large signatures, causing communication overhead. Hence, the latency overhead in DA is reduced using the Struve Function (SF).

Initially, centered on SF, the Dilithium private key (λ) is set. SF models and smooths the oscillatory and delay-dependent behaviors.

$$\lambda = u(q,i) + \left\lfloor \frac{R(f)}{1 + R(f)} \right\rfloor \tag{14}$$

Where, the hash value is signified as (u), the SF decay order is symbolized as (R), and the Dilithium normalization factor is denoted as (f). Now, grounded on the public key (s) and (λ), the signature $(\Im)$ is produced at the time of slice ID initialization.

$$\Im = (q, i, \lambda, s) \tag{15}$$

Next, to evade tampering, $(\Im)$ is stored in the blockchain. The stored $(\Im)$ is used for verification. Then, the trust of the slice is evaluated.

3.4 Trust Assessment

Then, grounded on throughput (H), packet delivery ratio (F), and packet delay (I), the trust score (μ) of (Z) is evaluated.

$$\mu = k\left[H + F + (1 - I)\right] \tag{16}$$

$$\mu \rightarrow \begin{cases} if & \mu \geq 0.85 & then\ D_1 \\ if & 0.70 \leq \mu < 0.85 & then\ D_2 \\ if & \mu < 0.70 & then\ D_3 \end{cases} \tag{17}$$

Where, the weight value of each parameter is signified as (k) and the deployment status of the network slices, like deployable, conditional deployable, and non-deployable, is symbolized as (D_1, D_2, D_3), respectively.

3.5 Network Slice Deployment

Then, the HGGaO is utilized to assign correct network slices for tenant data storage. Here, for network slice deployment, the Gazelle Optimization (GaO) that effectively balances the global (exploration) and local (exploitation) search is chosen. Nevertheless, GaO converges prematurely to suboptimal solutions. So, in the GaO's exploration phase, the Hyperbolic Growth (HG) function is utilized.

Firstly, with respect to the upper bound (ub^Z), lower bound (lb^Z), and a random value (r), the position (Z^q) of the Gazelle (network slice) is determined.

$$Z^q = \left[r \times \left(ub^Z - lb^Z\right)\right] + lb^Z \tag{18}$$

Further, to determine the optimal solution, the fitness function (ψ) is set. Here, to articulate (ψ), (μ) and slice score (x) are utilized.

$$\psi = (c \times \mu) + \left[(1 - c) \times (x)\right] \tag{19}$$

$$x = \left[\gamma + (1 - \eta)\right] Z \tag{20}$$

Here, trust weight is signified as (c). Next, the position of (Z^q) is updated. At the time of exploration, the Gazelle reacts to the danger by flicking its tail. Grounded on exploration, the new position (Z^{q+1}) is expressed as,

$$Z^{q+1} = Z^q + J \times \left[r \times \left(Z^q\right)\right] \tag{21}$$

$$J = 1/(1 + a \times L) \tag{22}$$

Where, HG function that maintains strong exploration for gradual shifting is signified as (J), and the hyperbolic decay factor and present iteration are symbolized as (a, L), correspondingly. Then, the Gazelle grazes peacefully in the exploitation phase. The position $\left(Z^{q+2}\right)$ with respect to exploitation is mathematically represented as,

$$Z^{q+2} = Z^{q+1} + r\left[Z^{q+1} - Z^{q+1}(L)\right] \tag{23}$$

Thus, by applying (ψ), the deployment (D) is performed by the orchestrator.

Pseudo-code for HGGaO

Input: NSP (Z), Trust Score (μ)

Output: Network Slice Deployment (D)

Begin

For (Z)

Initialize $Z^q = \left[r \times \left(ub^Z - lb^Z\right)\right] + lb^Z$

Determine slice score $x = [\gamma + (1-\eta)]Z$

Set fitness $\psi = (c \times \mu) + [(1-c) \times (x)]$

While (iteration = L)

Estimate HG function $J = 1/(1 + a \times L)$

Find exploration position $Z^{q+1} = Z^q + J \times \left[r \times \left(Z^q\right)\right]$

Evaluate exploitation position $Z^{q+2} = Z^{q+1} + r\left[Z^{q+1} - Z^{q+1}(L)\right]$

Apply fitness (ψ)

Obtain (D)

End while

End for

Return (D)

End

Hence, scalable and secure network slice deployment is guaranteed, and the tenant data is secured.

3.6 Tenant Data Security

Then, the CKBNAChaCha20 is used to secure the tenant (T)'s data (M). Here, for data security, the ChaCha20 stream cipher model that transforms the tenant data into a keystream through eXtreme OR (XOR) operation is chosen. Nevertheless, an attacker who attains the key could decrypt all tenant data. Hence, in ChaCha20, the key $\left(h^{\text{shared}}\right)$ from CKBNA (Section 3.1) is used.

Firstly, centered on a 96-bit nonce (p), block index (B), and the key $\left(h^{\text{shared}}\right)$ from CKBNA, the key stream (ξ) is assessed.

$$\xi = B\left[p, h^{\text{shared}}\right] \tag{24}$$

Further, the data (M) is secured with respect to (ξ) for evading quantum and other adversaries. The secured data $\left(M^{\text{secured}}\right)$ is articulated as,

$$M^{\text{secured}} = M + \xi \tag{25}$$

Hence, the tenant's data is secured. Then, the signature is verified.

3.7 Signature Verification

Here, the signature verification is performed utilizing SFDA for configuring the cloud and the slice and evading the cross-cloud data leakage. For verification, the signature $(\Im)$ that is produced in Section 3.3.1 is utilized. Before storing $\left(M^{\text{secured}}\right)$ into (Z), the verification is conducted.

$$X = \begin{cases} \ddot{X} & if\left[\Im(Z) = \Im(C)\right] \\ X''' & if\left[\Im(Z) \neq \Im(C)\right] \end{cases} \tag{26}$$

The signature verification (X) exhibits that if $(\Im)$ of the slice (Z) and cloud (C) are equal, then the signature is verified $\left(\ddot{X}\right)$. If $(\Im)$ of (Z) and (C) are not equal, then the signature is not verified (X'''). And, if (X''') is attained, then the storage of $\left(M^{\text{secured}}\right)$ is obstructed. If $\left(\ddot{X}\right)$ is obtained, then the storage takes place in (Z). Further, by utilizing the shared keys and the generated signature, the data needs to be decrypted.

3.8 Blockchain

Lastly, in the blockchain, all the transactions, such as tenant registration, NSP, slice deployment, and authenticated data storage, are recorded. Hence, an immutable, transparent, and tamper-resistant audit is carried out. Thus, in a multi-cloud network, the proposed system secures the network slicing. The performance validation is explained in the following section.

4. RESULTS AND DISCUSSION

Here, the proposed models' performance is validated and contrasted with existing models. The proposed model is implemented on the PYTHON platform.

4.1 Dataset Description

The "5G Network Slicing Dataset" is used for NSP. The dataset link is provided in the reference section. A total of 63167 data are available. Among those, 80% (50533 data) are used for training the NSPM, whereas 20% (12634 data) are utilized for testing the NSPM.

4.2 Performance Evaluation

Here, the proposed models' performance is analogized with the extant methodologies.

Table 1: Comparative Analysis of CKBNA

Techniques	KET (ms)	SKER (%)	Latency (ms)
Proposed CKBNA	1039	2.467	1842
CKA	3849	5.832	4728
NHA	5721	8.491	7461
FrodoKEM	7942	11.942	10472
ECDH	9854	14.623	13527

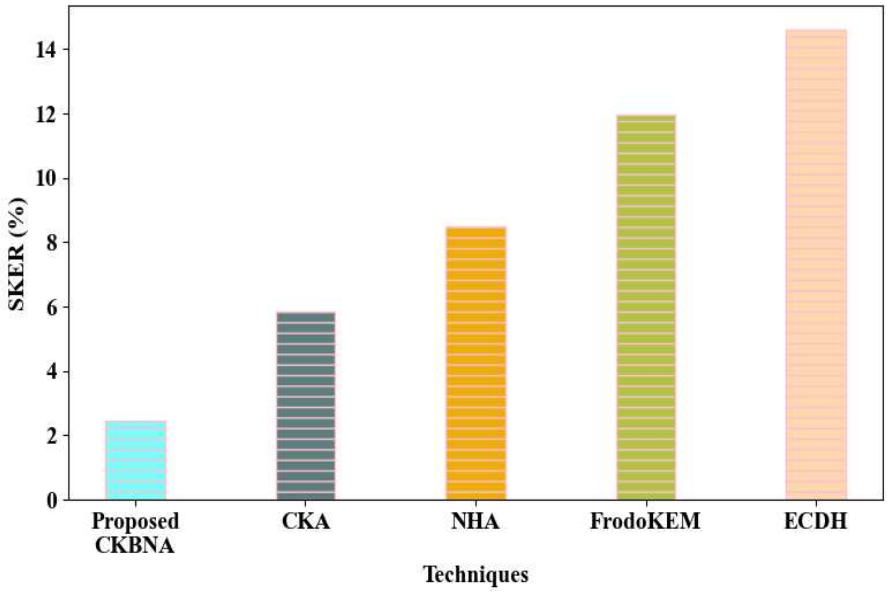


Figure 3: Graphical Comparison of CKBNA

Table 1 and Figure 3 illustrate that the SKE is carried out by the proposed CKBNA with a 1039ms Key Establishment Time (KET), a 2.467% SKER, and a 1842ms latency. Yet, an average KET, SKER, and latency of 6842ms, 10.22%, and 9047ms are obtained by the existing CKA, NewHope Algorithm (NHA), Frodo-Key Encapsulation Mechanism (FrodoKEM), and ECDH, correspondingly. Thus, the performance of the proposed SKE model is improved by the BN function-based distribution.

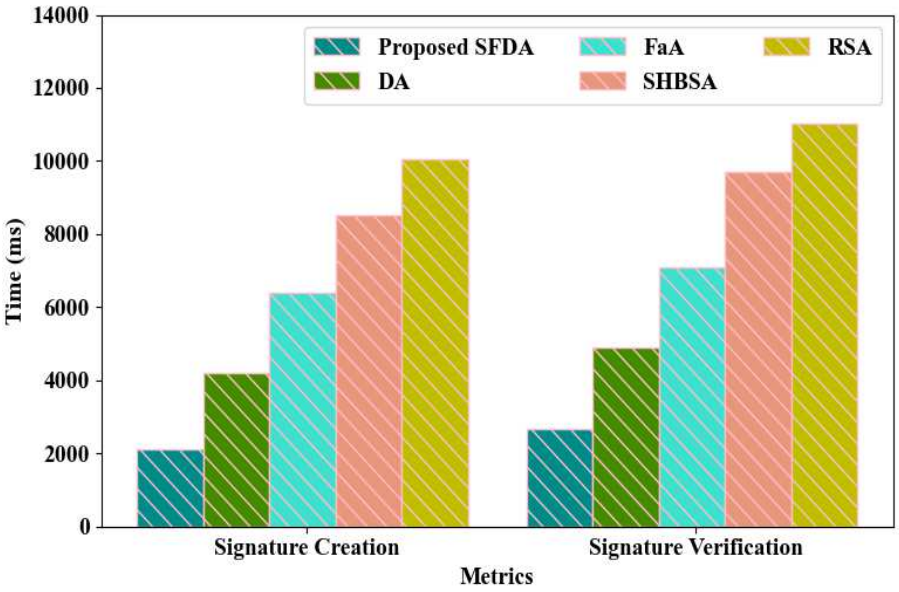


Figure 4: Comparative Analysis of SFDA

By using SF, the proposed technique tackles the computational overhead. Hence, Figure 4 shows that the network slice and the respective cloud are authenticated by the proposed SFDA with a 2104ms Signature Creation Time (SCT) and 2673ms Signature Verification Time (SVT). But, an SCT of 4215ms, 6381ms, 8525ms, and 10038ms, and SVT of 4902ms, 7093ms, 9693ms, and 11027ms are attained by the traditional methods, such as DA, Falcon Algorithm (FaA), Stateless Hash-Based Signature Algorithm (SHBSA), and Rivest-Shamir-Adleman (RSA), respectively. Hence, the proposed SFDA performed superiorly when contrasted with extant methodologies.

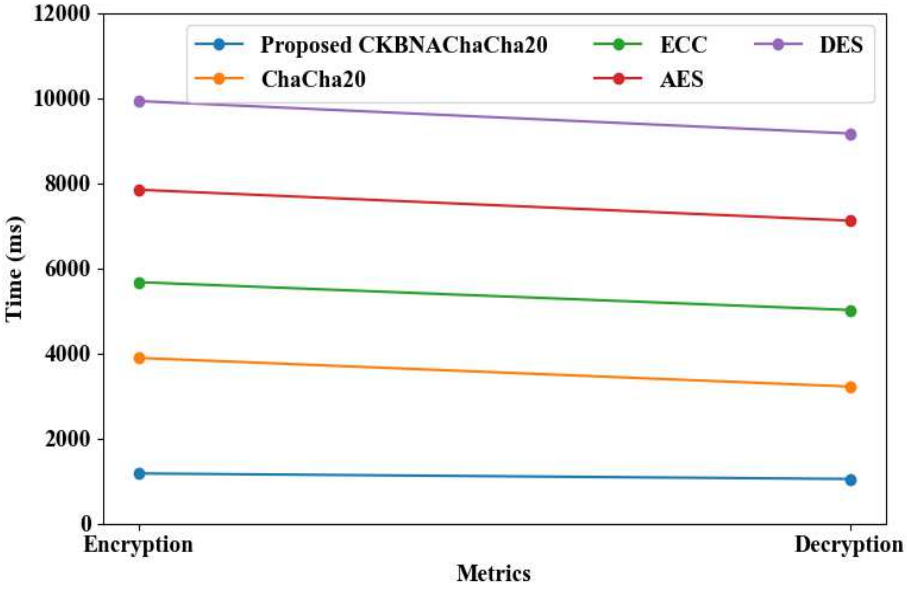


Figure 5: Comparative Analysis of CKBNAChaCha20

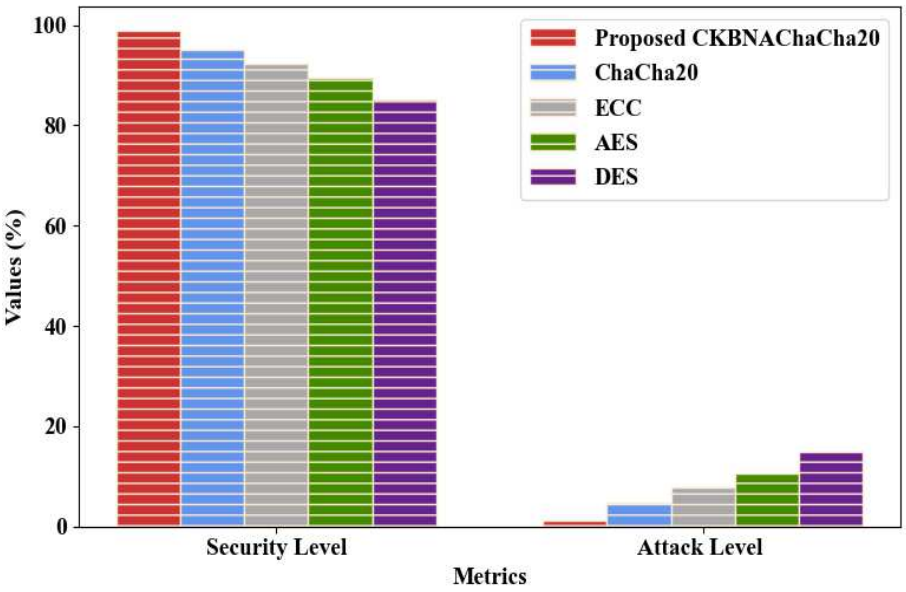


Figure 6: Graphical Comparison regarding Tenant Data Security

It is depicted in Figures 5 and 6 that the proposed CKBNAChaCha20 secures the tenant's data with a 1178ms encryption time, 1048ms decryption time, 98.78% security level, and 1.22% attack level. Nevertheless, an average encryption time of 6833ms, decryption time of 6129ms, security level of 90.48%, and attack level of 9.53% are attained by the prevailing ChaCha20, Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), and Data Encryption Standard (DES). Thus, the proposed data security process is enhanced by the CKBNA-based key.

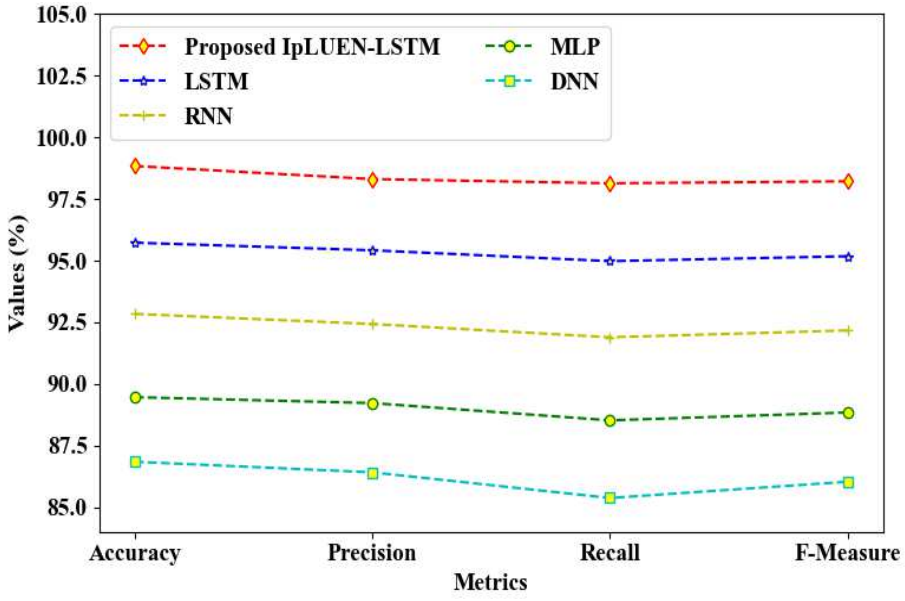


Figure 7: Comparative Analysis of IpLUEN-LSTM

In the proposed classifier, the IpLU activation function and EN are utilized. Hence, Figure 7 depicts that the network slice is predicted by the proposed IpLUEN-LSTM with an accuracy, precision, recall, and F-Measure of 98.84%, 98.31%, 98.14%, and 98.22%, respectively. Yet, when contrasted to the proposed methodology, lower accuracy, precision, recall, and F-Measure are attained by the extant LSTM, Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP), and Deep Neural Network (DNN). This proves the proposed classifier's improvement.

Table 2: Comparative Analysis of HGGaO

Models	DSR (%)				
	50 Nodes	100 Nodes	150 Nodes	200 Nodes	250 Nodes
Proposed HGGaO	99.21	99.17	99.05	98.93	98.82
GaO	96.93	96.57	96.23	95.84	95.57
BEO	93.78	93.21	92.81	92.55	92.04
RPO	90.63	90.25	89.78	89.43	89.17
PSO	87.52	87.26	86.95	86.63	86.24

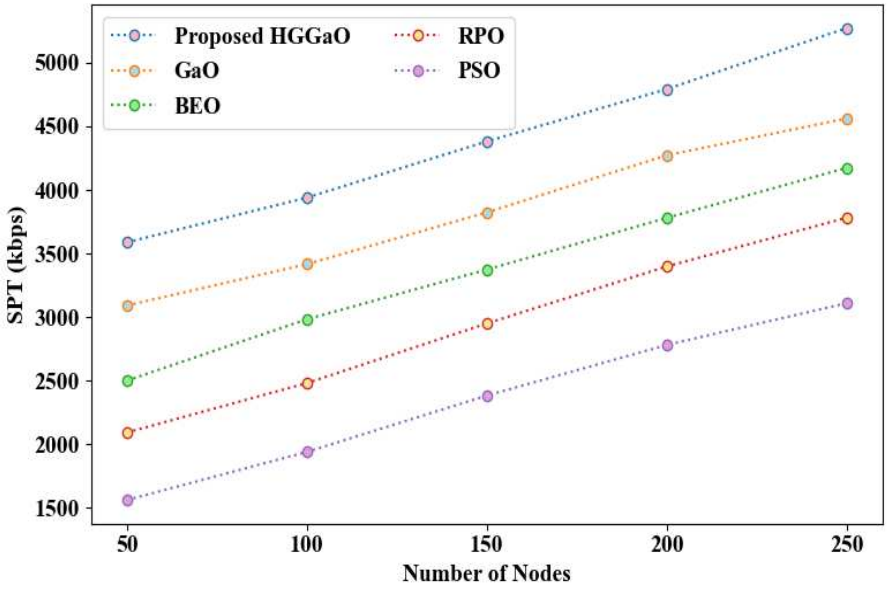


Figure 8: Graphical Comparison of HGGaO

By utilizing the HG function, the exploration phase with respect to network slice deployment is improved. It is depicted in Table 2 and Figure 8 that the proposed HGGaO attains a Deployment Success Rate (DSR) of 99.21%, 99.17%, 99.05%, 98.93%, and 98.82%, and Slice Provisioning Throughput (SPT) of 3589kbps, 3940kbps, 4382kbps, 4793kbps, and 5273kbps, for 50, 100, 150, 200, and 250 nodes, correspondingly. However, when compared to the proposed optimizer, lower DSR and lower SPT are obtained by the conventional GaO, Bald Eagle Optimization

(BEO), Red Panda Optimization (RPO), and Particle Swarm Optimization (PSO). Thus, the proposed model performed better than the prevailing models.

Table 3: Existing Works Comparison

Study	Objectives	Methods	Advantages	Drawbacks
Proposed Work	SNS in multi-cloud.	CKBNA, SFDA, IpLUEN-LSTM, HGGaO, and CKBNAChaCha20	Scalable and SNS were obtained.	Intrusion was not identified.
(Nong & Liang, 2025)	Allocation of network slice.	Exponential function scheduling	Achieved higher throughput.	Secure slicing was not maintained.
(Andreou et al., 2025)	Network slices security.	Deep Reinforcement Learning (DRL)	Maintained high network performance.	Learning efficacy was degraded.
(Venkatapathy et al., 2024)	5G network orchestration.	Dual-Slice Isolation and Management Strategy (D-SIMS)	Resources were allocated efficiently.	Dynamic orchestration was missing.
(Kumar et al., 2025)	5G network security.	AES and DES	Minimised encryption/ decryption time.	Receptive to quantum attacks.
(Min et al., 2024)	Enhancing 5G network slicing.	KMeans	Slicing was provisioned.	Could not manage a large-scale network.

A comparison of the proposed work and prevailing works is shown in Table 3. SKE, NSP, and security of the tenant's data are applied by the proposed framework. Hence, the cross-cloud data leakage is evaded. Nevertheless, the prevailing works have various limitations. These limitations are tackled by the proposed work. This proves the proposed work's improved performance.

5. CONCLUSION

SNS for a multi-cloud environment was performed in this paper. Initially, the tenant registered in the cloud. Further, using CKBNA, SKE was done, which achieved a SKER of 2.467%. Thereafter, the NSP was carried out by utilizing IpLUEN-LSTM, which attained 98.84% accuracy. Furthermore, the slice ID was initiated, and the trust score was evaluated. Subsequently, the signature was generated by SFDA within 2104ms. Next, using HGGaO, the node slice was deployed with 99.21% DSR for 50 nodes. After that, the tenant's data was secured by utilizing CKBNAChaCha20 with a security level of 98.78%. Eventually, by using SFDA, the signature was verified within 2673ms. Further, in the multi-cloud, the data was stored. Along with that, all the transactions were recorded in the blockchain. Hence, the network slicing was effectively secured by the proposed work.

Practical Implication: An end-to-end security for network slicing, which is appropriate for 5G/6G networks, cloud service providers, smart cities, telecom operators, and so on, is provided by the proposed framework.

Future Scope: Intrusion detection will also be incorporated in the future to monitor the slices for malicious activities.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent of publication: Not applicable.

Availability of data and materials: Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests: The authors declare that they have no competing interests

Funding: This work has no funding resource.

Author's contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [*1]**Corresponding Author,** [2]**Second Author.** The first draft of the manuscript was written by [*1]**Corresponding Author** and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Acknowledgement: We thank the anonymous referees for their useful suggestions.

REFERENCES:

Dataset Link: https://www.kaggle.com/datasets/puspakmeher/networkslicing

Abdellatif, A. A., Abo-Eleneen, A., Mohamed, A., Erbad, A., Navkar, N. V., & Guizani, M. (2023). Intelligent-Slicing: An AI-Assisted Network Slicing Framework for 5G-and-Beyond Networks. *IEEE Transactions on Network and Service Management*, *20*(2), 1024–1039. https://doi.org/10.1109/TNSM.2023.3274236

Alnaim, A. K. (2024). Securing 5G virtual networks: a critical analysis of SDN, NFV, and network slicing security. *International Journal of Information Security*, *23*, 3569–3589. https://doi.org/10.1007/s10207-024-00900-5

Andreou, A., Mavromoustakis, C. X., Markakis, E., Bourdena, A., & Mastorakis, G. (2025). Enhancing network slice security with Deep Reinforcement Learning and Moving Target Defense strategies. *Discover Internet of Things*, *5*(1), 1–17. https://doi.org/10.1007/s43926-

025-00161-1

Benlloch-Caballero, P., Matencio-Escolar, A., Bernabe, J. B., Skarmeta, A., Wang, Q., & Alcaraz-Calero, J. M. (2025). E2E Network Slicing for Enhanced Cybersecurity, Orchestration, Automation and Response in 5G/6G: The RIGOUROUS Approach. *Journal of Network and Systems Management*, *34*(1), 1–36. https://doi.org/10.1007/s10922-025-09999-w

Benzaid, C., Taleb, T., & Song, J. (2022). AI-Based Autonomic and Scalable Security Management Architecture for Secure Network Slicing in B5G. *IEEE Network*, *36*(6), 165–174. https://doi.org/10.1109/MNET.104.2100495

Bharot, N., Mehta, N., Breslin, J. G., & Verma, P. (2025). Cloudlock: secure data sharing using a hybrid cryptosystem in multi-cloud data storage. *Cluster Computing*, *28*(7), 1–21. https://doi.org/10.1007/s10586-025-05433-7

Cui, Q., You, X., Wei, N., Nan, G., Zhang, X., Zhang, J., Lyu, X., Ai, M., Tao, X., Feng, Z., Zhang, P., Wu, Q., Tao, M., Huang, Y., Huang, C., Liu, G., Peng, C., Pan, Z., Sun, T., … Yuen, C. (2025). Overview of AI and communication for 6G network: fundamentals, challenges, and future research opportunities. *Science China Information Sciences*, *68*(7), 1–61. https://doi.org/10.1007/s11432-024-4337-1

Diallo, E. hacen, Agha, K. Al, & Martin, S. (2025). TRADE-5G: A blockchain-based transparent and secure resource exchange for 5G network slicing. *Blockchain: Research and Applications*, *6*(1), 100246. https://doi.org/10.1016/j.bcra.2024.100246

Etienne, I. T., Firdaus, M., Latt, C. N. Z., Noh, S., & Rhee, K. H. (2024). Hyperledger Fabric-Powered Network Slicing Handover Authentication. *Information Systems Frontiers*, 1–19. https://doi.org/10.1007/s10796-024-10564-0

Irshad, R. R., Hussain, S., Hussain, I., Nasir, J. A., Zeb, A., Alalayah, K. M., Alattab, A. A., Yousif, A., & Alwayle, I. M. (2023). IoT-Enabled Secure and Scalable Cloud Architecture for Multi-User Systems: A Hybrid Post-Quantum Cryptographic and Blockchain-Based Approach Toward a Trustworthy Cloud Computing. *IEEE Access*, *11*, 105479–105498. https://doi.org/10.1109/ACCESS.2023.3318755

Joshi, H. (2025). Emerging Technologies Driving Zero Trust Maturity Across Industries. *IEEE Open Journal of the Computer Society*, *6*, 25–36. https://doi.org/10.1109/OJCS.2024.3505056

Kumar, A., Singh, P., Kamble, D. P., & Singh, I. (2025). Hybrid cryptographic approach for strengthening IoT and 5G/B5G network security. *Scientific Reports*, *15*(1), 1–20. https://doi.org/10.1038/s41598-025-21861-2

Min, Z., Gokhale, S., Shekhar, S., Mahmoudi, C., Kang, Z., Barve, Y., & Gokhale, A. (2024). Enhancing 5G network slicing for IoT traffic with a novel clustering framework. *Pervasive and Mobile Computing*, *104*, 101974. https://doi.org/10.1016/j.pmcj.2024.101974

Nong, X., & Liang, X. (2025). Scheduling allocation in 5G slicing networks utilizing weighted

exponential and logarithmic functions to improve QoS. *Scientific Reports*, *15*(1), 1–25. https://doi.org/10.1038/s41598-025-17385-4

Tabatabaei, F., Mangues-Bafalluy, J., Khalili, H., Kahvazadeh, S., & Requena-Esteso, M. (2025). Optimizing CNF Placement for Latency and Resource Efficiency in Multi-Cluster Cloud-Edge Federations. *IEEE Open Journal of the Communications Society*, *6*, 8376–8388. https://doi.org/10.1109/OJCOMS.2025.3614745

Venkatapathy, S., Srinivasan, T., Lee, O. S., Jayaraman, R., Jo, H. G., & Ra, I. H. (2024). Slice-aware 5G network orchestration framework based on dual-slice isolation and management strategy (D-SIMS). *Scientific Reports*, *14*(1), 1–21. https://doi.org/10.1038/s41598-024-68892-9

Wang, W., Liu, R., & Cheng, S. (2024). Privacy protection of communication networks using fully homomorphic encryption based on network slicing and attributes. *Scientific Reports*, *14*(1), 1–18. https://doi.org/10.1038/s41598-024-69501-5

Zamfirescu, C., Iugulescu, R., Crăciunescu, R., Vulpe, A., Li, F. Y., & Halunga, S. (2024). Network slice allocation for 5G V2X networks: A case study from framework to implementation and performance assessment. *Vehicular Communications*, *45*, 1–12. https://doi.org/10.1016/j.vehcom.2023.100691

Zeydan, E., Baranda, J., & Mangues-Bafalluy, J. (2022). Post-Quantum Blockchain-Based Secure Service Orchestration in Multi-Cloud Networks. *IEEE Access*, *10*, 129520–129530. https://doi.org/10.1109/ACCESS.2022.3228823

Zeydan, E., Blanco, L., Barrachina-Munoz, S., Rezazadeh, F., Vettori, L., & Mangues, J. (2023). A Marketplace Solution for Distributed Network Management and Orchestration of Slices. *3rd International Workshop on Analytics for Service and Application Management (AnServApp 2023)*, 1–6. https://doi.org/10.23919/CNSM59352.2023.10327832