HKC-MBLT: Enhancing SaaS Transaction Security with Hybrid Cryptography and Memory-Based Lightweight Tokenization for IoT-Enabled Cyber-Physical Systems

Kennedy Chinedu Okafor[1,2], Omowunmi Mary Longe[2], Michael Obinna Ezeja[3], Ikechukwu Ignatius Ayogu[4], Kelvin Anoh[5], Bamidele Adebisi[1]

[1]School of Engineering, Manchester Metropolitan University, M15 6BH, Manchester, UK.
[2]Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
[3]Department of Electronic and Computer Engineering, University of Nigeria, Nsukka
[4]Department of Computer Science, Federal University of Technology, Owerri, Nigeria, PMB 1526
[5]School of Engineering, University of Chichester, Bognor Regis, PO21 1HR, U.K
Email: kennedy.okafor@mmu.ac.uk, omowunmil@uj.ac.za, obinna.ezeja@unn.edu.ng, ignatius.ayogu@futo.edu.ng k.anoh@chi.ac.uk, b.adebisi@mmu.ac.uk

Corresponding Author: Kennedy Chinedu Okafor[1,2] kennedy.okafor@mmu.ac.uk

Abstract

As the integration of Internet of Things (IoT) devices into Software-as-a-Service (SaaS) platforms expands, security concerns in digital environments have grown significantly. Traditional public key cryptographic schemes, including Diffie-Hellman (DH) and Elliptic Curve Cryptography (ECC), face high computational demands and key management vulnerabilities, which are exacerbated in resource-constrained environments like IoT devices and edge modules. To address these challenges, we propose a Hybrid Key Cryptographic Engine (HKCE), coupled with a Memory-Based Lightweight Tokenization (MBLT) approach for enhanced access authentication in Payment Transaction Systems (PTS). This hybrid cryptographic framework optimises encryption processes, mitigates bandwidth vulnerabilities, and offers quantum-resistant resilience against emerging cryptographic threats. The performance of the proposed HKCE-MBLT solution is benchmarked against traditional ECC-Key Scattering Schemes (ECC-KSS), demonstrating a significant reduction in computational overhead (25% compared to 75% for ECC-KSS), higher throughput (78.57% compared to 21.43% for ECC-KSS), and lower bandwidth vulnerability, while maintaining the integrity, confidentiality, and availability of transactions. The proposed solution provides a scalable, efficient, and secure framework that ensures privacy and trust in IoT-enabled SaaS systems, positioning it as a robust alternative for securing payment card systems and other sensitive applications in the evolving digital landscape.

Keywords: Asymmetric cryptography, SaaS transactional systems, IoT-enabled devices, hybrid key cryptography, lightweight authentication.

1. Introduction

Global spending on public cloud ecosystems is projected to reach $725 billion by the end of 2024, with over 75% of enterprises relying heavily on cloud-powered platforms to conduct business operations [1]. Hyperscalers such as AWS, Microsoft, and Google dominate the enterprise market, extending to the Internet of Things and cyberphysical systems. These entities also offer cost-efficient solutions that reduce the total cost of ownership (TCO) of enterprise security infrastructure [2]. With the increasing adoption of generative artificial intelligence (GenAI), IoT, and microservices application program interfaces (API) in the SaaS domain, security has become more critical than ever [3]. These platforms operate within complex layered security ecosystems, including identity management and firewalls, making SaaS solutions essential for modern enterprises [4].

To date, online transactions rely heavily on integrated SaaS and cyber-physical devices, such as point-of-sale terminals, mobile payment apps, and IoT enabled payment systems [5]. As SaaS transactional systems increasingly integrate IoT devices, they encounter a complex evolution in attack dynamics and vulnerabilities, driven by the accelerating digital transformation in e-commerce. The inclusion of IoT devices significantly expands the attack surface, introducing new vectors for cyber threats. These interconnected devices create opportunities for sophisticated attacks, including data breaches, unauthorized access, and asymmetric cyberattacks, which can undermine the integrity and confidentiality of critical transactional data. As a result, e-commerce SaaS platforms must adopt advanced security strategies to address the unique challenges posed by the convergence of cloud services and IoT technology [6].

Moreover, IoT devices often have resource limitations that make them particularly vulnerable to exploitation, complicating defense measures and escalating costs, especially when relying on traditional cryptographic approaches like Elliptic Curve Cryptography (ECC) [7]. A key challenge in securing SaaS e-commerce systems lies in managing the computational overhead and vulnerabilities associated with conventional encryption methods, which can adversely impact network performance [8].

In context, the Public Key Infrastructure (PKI) has been widely adopted to address these issues, employing asymmetric-key cryptography for secure encryption and decryption of digital services, particularly within cloud environments. Operationally, PKI cryptographic-keys are linked to a digital certificate, thereby authenticating any interface, device, or user initiating the communication digitally. PKI solutions, although effective in ensuring authentication and access control, often lack the lightweight computational framework necessary for efficient performance in SaaS environments. Additionally, current methods for encrypting traffic and managing key exchanges, such as Elliptic Curve Diffie-Hellman (ECDH), are computationally demanding and introduce bandwidth strain, particularly in IoT-enabled systems [9], [10].

A recent study on SaaS-based e-commerce systems highlighted critical challenges in traditional Public Key Cryptography (PKC), such as vulnerability to brute-force attacks, private key failures, and susceptibility to middleman attacks [11, 12]. These were compounded by the absence of lightweight computations. This means that resource consumption, such as CPU utilization, memory drain, and energy consumption are rarely optimized. While asymmetric PKC uses encryption schemes such as Rivest–Shamir–Adleman (RSA) [13], Diffie-Hellman, and elliptic Curve cryptography (ECC) [14] to secure communications, the computational demands remain significant. Issues such as outdated signature generation (e.g., SHA-256 with RSA) and the lack of forward secrecy further highlight the need for a lightweight encryption solution to efficiently secure e-commerce transactions. In addition, the absence of forward secrecy and weak encryption profiles in many PKI schemes further complicates matters. With lightweight encryption schemes, it is feasible to mitigate vulnerabilities in SaaS e-commerce systems, thereby reducing bandwidth strain and computational load.

To overcome these challenges, this study proposes a Hybrid Key Cryptographic Engine (HKCE) combined with Memory-Based Lightweight Tokenization (MBLT). This approach optimizes the encryption processes and minimizes the computational demands of securing SaaS transactional systems, offering a lightweight alternative to existing cryptographic techniques. By integrating probabilistic models with the Lightweight Key Encryption Mechanism (LKEM), this novel solution enhances the security of IoT-enabled e-commerce systems while reducing encryption overhead, improving throughput, and mitigating bandwidth vulnerabilities.

This study introduces a novel probabilistic lightweight algorithm for enhancing Electronic Card Transaction Systems (ECTS) security. The algorithm generates optimal density distributions using the MBLT, thereby overcoming the computational obstacles associated with legacy PKI systems. To the best of our knowledge, this study is the first to address ECTS computational probability in conjunction with LKEM. The approach first employs the Poisson Bayesian probability scheme and integrates it with the LKEM to optimizes the ECTS efficiency. The combination of HKCE and MBLT for IoT-enabled SaaS transactional systems is a novel contribution. The introduced MBLT will improve security in the payment card industry data security standards (PCI-DSS). The MBLT-Key Encapsulation Mechanism (KEM) is shown to complete cryptographic key distribution and optimize payment gateways. The implication is the potential enhancement of encryption and decryption processes in ECTS. Our evaluation effort is focused on Quality of Service (QoS) metrics such as computational overhead, throughput, latency, and vulnerability index, ensuring only authorized access to sensitive payment data.

A summary of the contributions is as follows:

- Achieved a Lightweight Poisson Probability Model that reduces the attack surface by determining the prior probability of customer participation in SaaS transactions.
- Derived a Computational Associative Memory Algorithm that leverages Bayesian techniques to optimize solutions using prior and posterior probability distributions.
- Proposed an Unbiased Uniform Estimator that assesses service accessibility within ECTS.
- Performance Validation: The MBLT KEM is compared against the ECC key scattering AES-CBC-mode technique using metrics such as computational overhead, throughput, execution time, and vulnerability index.

The remainder of this paper is organised as follows. Related work on e-commerce cryptographic schemes is presented in section 2. Section 3 introduces E-Commerce Reference architecture and its attributes. Section 4 describes the system model and cryptographic derivation. Section 5 presents the MBLT public key encryption (PKE) preliminaries. The performance evaluation is discussed in Section 6. Finally, Section 7 concludes the paper and provides directions for future research.

2. Related Works

In this section, we use Table 1 to summarize the existing efforts in various PKI security techniques particularly Diffie-Hellman, and elliptic curve cryptography. Both schemes are mostly used in cloud SaaS applications and IoT-related ecosystems.

Table 1. Recent SaaS Asymmetric Cryptographic efforts (Diffie-Hellman, and elliptic curve cryptography).

Ref	Year	ECTS Type	Security Mitigation Technology	Limitations	Application Domain
[17]	2024	IoMT-based healthcare system	ECC, one-way hash function, XOR operation, offline and online authentication	Potential computation and communication overhead	IoMT device authentication in healthcare networks
[18]	2022	Online public key cryptography, (PKC)	Elliptic Curve, Diffie-Hellman Key Exchange (GSAKA-ECDHKE)	Computational cost and Size	Mobile applications
[19]	2022	Online public key cryptography (i.e., (PKC-DDMIA)	Elliptic curve discrete log problem (ECDLP) & Elliptic curve computational Diffie–Hellman problem (ECDH	Computational cost & open to attack vectors	Hospital blockchain e-health application
[20]	2022	Online public key cryptography (PKC)	Post-quantum cryptography (PQC)	Survey post-quantum signature schemes lacking in discussions on computational. schemes	IoT-based services
[21]	2024	Cryptocurrency wallet systems	Elliptic-curve cryptography (ECC), seed phrase encryption and splitting	Vulnerability if a portion of the seed phrase is known	Cryptocurrency transaction security and wallet protection
[22]	2022	Cloud-based access control systems	Policy Authenticable Attribute-Based Encryption (PA-ABE)	Vulnerability to fake policy attacks	Applications requiring strict access control and data confidentiality
[23]	2023	Cloud-based location services	Secure R-tree (SR-tree) index, permutation, perturbation, and dominance protocol.	High computational costs for ciphertext processing	Secure location-based skyline queries in cloud SaaS systems
[24]	2024	Cyber-physical power system transactions	Dynamical Pseudonym Self-Generation Mechanism (DPSGM), ECC, certificateless cryptography	Pseudonym update limitations in existing schemes; computation and communication overheads	Resource-limited smart terminals (STs) in power systems
[25]	2024	IoMT-based healthcare transactional system	SaaS-based Intrusion Detection System (IDS)with particle swarm optimization (PSO), ML/DL+ Shapley additive explanations (SHAPs)	Resource-constrained devices; computational limitations	SaaS Healthcare IoMT networks; data privacy and network integrity
[26]	2023	Cloud-based time series query system	Symmetric Homomorphic Encryption (SHE), secure sort protocol, dominance check tree	Potential data utility and query efficiency loss	Privacy-preserving skyline queries for time series data in the cloud
[27]	2024	Cloud-based systems integrating IoT	Scalable and Secure Cloud Architecture (SSCA), MBRA, Post Quantum Cryptography (PQC), blockchain	Complexity in managing decentralized cloud nodes and algorithms	Collaborative computing for multi-user access in cloud environments with IoT
[28]	2023	Systems employing Diffie-Hellman (DH) key exchange (e.g., TLS, SSH)	Recommend transitioning to Elliptic Curve Diffie-Hellman	Vulnerable to D(HE)at DoS attack; affected by imple-	General-purpose cryptographic libraries, impact-

			(ECDH) for better efficiency.	mentation flaws allowing large private keys	ing server configurations worldwide
[29]	2023	IoT environments using Diffie-Hellman (DH) key exchange	Asymmetric computing cryptosystem; ACKE protocol	Traditional cryptographic protocols have high overhead; and resource constraints in terminal devices.	Internet of Things (IoT)
[30]	2024	Public-key steganography	Public-key steganography method based on elliptic curve cryptography and generative models	Challenges with key agreement, key updating, and user expansion in practical settings	Secure communication and data hiding
[31]	2024	Industrial Internet of Things (IIoT)	Identity-based authenticated key exchange protocol using lattice cryptography.	Vulnerable to quantum cryptanalysis; traditional DH methods have overhead	Open wireless communication in IIoT
[32]	2023	Internet of Things (IoT)	CS-LAKA: a secure lightweight authenticated key agreement (AKA) protocol	Inefficiency of public-key operations for resource-constrained devices; symmetric protocols are vulnerable to attacks	Secure communication in IoT environments
[33]	2023	Group key establishment protocols	Authenticated distributed group key agreement protocol (ADGKAP) using Elliptic Curve Secret Sharing Scheme (ECSSS)	Limited existing literature on ECSSS in distributed environments	Resource-constrained Internet of Things (IoT) applications
[34]	2023	Cloud-based outsourced storage systems	'Isogeny'-based Blinded Key Encapsulation Mechanism (BKEM) and forward secure Offline Assisted Group Key Exchange (OAGKE) protocol.	Previous BKEM constructions are vulnerable to quantum attacks	Secure file storage and sharing in offline collaborative environments
[35]	2023	Internet of Things (IoT) networks	Key agreement and authentication mechanism based on Elliptic-Curve Diffie-Hellman (ECDH)	RPL protocol vulnerabilities; limited resources of IoT nodes; symmetric key encryption weaknesses	Secure data exchange and authentication in IoT networks

2.1 E-commerce Transaction Attacks

Within the various E-commerce Transaction (ECT) ecosystems, the use of PKC offers security to emerging threats and attacks such as backdoor [36], man-in-the-middle attacks [37], replay attacks [38], key distribution complexities in payment systems [39], point-of-transaction (POT) theft [40], eavesdropping [41], phishing [42], Distributed Denial-of-Service (DDoS) [43, 44]. Although PKC is widely adopted, its limitations are summarized as follows:

- Most works failed to discuss key management efforts especially once the private keys were compromised by an attack vector that could affect security ecosystems.
- Existing studies have not explored hybrid schemes that can mitigate quantum signatures/attack-resistant algorithms.
- In general, PKC schemes including message encryption, decryption, and signature verification are both resource and computationally intensive in relation to the symmetric security approach. This makes it unattractive for resource-limited settings such as edge/IoT modules.

3. System Model

In this subsection, a robust encryption system that leverages a combination of a symmetric key and a lightweight encryption algorithm is proposed. This system ensures the confidentiality of messages during transmission, facilitating swift and dependable transaction payment delivery, as depicted in Figure 1. Our optimisation problem based on the PKI has an objective function and its related constraints. A structured formulation is presented.

3.1. Optimisation Problem Formulation

- ***Objective Function***

Maximize the efficiency of transaction processing and encryption in a proposed Online Smart Malls and Retail Shops (OSMARS) environment while minimizing the time spent per transaction. The objective can be expressed as Equation (1)

$$Maximise\ \omega.\left(\frac{1}{T}\right) \tag{1}$$

Where

- ω represents the average number of transactions per customer
- T represents the average time spent by each customer on transactions

Constraints

1. **Transaction Capacity Constraint:** Each edge capacity must not exceed the flow rate, ensuring that the encrypted transactions remain within the limits of the available resources. This is represented as Equation (2)

$$0 \leq f(e) \leq c(e), \forall_e \epsilon E \tag{2}$$

2. **Flow Conservation Constraint:** For each vertex v except for source s and sink t, the incoming and outgoing flow must be balanced, Equation (3)

$$\sum_{e^+=v} f(e) = \sum_{e^-=v} f(e), \qquad \forall_u \neq s,t \tag{3}$$

3. **Transaction Processing Time:** The total time spent on transactions must not exceed a predefined threshold T_{Max}:
   $T \leq T_{Max}$:
4. **Anonymity Requirement:** This ensures complete anonymity of customer data through secure encryption methods.

   Probability of obtaining information: This is defined by the function in Equation (4):

$$P\left({}^{yA_i}/_{\theta A_i}\right) \leq \varepsilon \tag{4}$$

   Where ε represents a small, predefined threshold that ensures minimal leakage of information.

   The utilisation of cryptographic tools ensures the use of lightweight cryptographic methods, such as MBLT, during transaction encryption while maintaining the integrity and confidentiality of transactions.

Model variables:

- $f(e)$: Flow through edge e
- $C(e)$: Caaity edge e
- n_{trans}**:** Number of transactions per customer per time
- $Prop.n = \frac{n_{trans}}{N}$: Fraction of transactions to the total transactions.
- P_{prior}: prior probabilities for customer transaction
- $P_{posterior}$: posterior probabilities derived from the Bayesian model.

3.2. Notation and Problem Formulation

The preliminary computational model definitions, notations, and tools used in this study are presented in Table 2. The system characterization was formulated based on the highlighted notes. We intend to determine the optimal flow $f(e)$ in the Bayesian flow network $N\ (G, c, s, t)$ that maximizes the efficiency of transaction processing while adhering to the constraints above, thereby enhancing the overall performance of the OSMARS system. We now

use this structured optimization problem to develop the scenario required to implement our proposed cryptographic scheme within the PKI ecosystem.

Table 2: Cryptographic Computational Notations.

Symbol	**Explanation**
Cstm	Customer.
n-Trans	**N**umber of transactions per customer per time
t(Sec)	time spent on a transaction in seconds.
Prop.n (n/N)	fraction of transactions to the total transactions
Pr.of Prio	probability of priors.
joint prb	joint probabilities.
Posterior	posterior probabilities.
ω	the average number of transactions per customer
T	average time spent by each customer in the transaction
A_i	Customers
M	the first collecting point
N	the secondary collection point
Q	distributed Cloud Space
$\Lambda(t)\theta t^b$	Mean of PLP (power-law process)
$A_i \sim NHPP(\theta t)$	= the random variable A_i follows a non-homogenous Poisson process with parameter titer *t*
$\frac{d}{dt}\Lambda(t) = abt^{b-1}$	Intensity
$P(y_{A_i}/\theta_{A_i}$	the probability of obtaining information about A_i given that it previously had the A_i history
$(T_{n(x)})$	Statistic, which is also an estimator of the population θ
$E(T_{n(x)})$	Expected value of the estimator (mean of the estimator)
$E(T_{n(x)})$	θ is the population parameter, $T_{n(x)}$ is an unbiased estimator).
$T_{n(x)} = \frac{y(y-1)}{n^2}$	Uniform minimum variance unbiased estimator (UMVUE)
$\sum_{i=1}^{n} x_i = y$	Cryptographic Sufficient statistic
$f(e)$	Optimal flow in the Bayesian flow network
$C(e)$	Edge capacity
$N(G, c, s, t)$	Bayesian flow network adversarial source s and sink t
$A_{a1}, A_{a2}, \dots, A_{an}$	Tier 1 administrators
S_a	Tier 2 Super administrator
$C_{c1}, C_{c2}, \dots, C_{cn}$	Customers of ECTS cloud

3.3. Design and Implementation of OSMARS Cryptographic Architecture

- ***Two-Tier Administration Framework for OSMARS***

To optimize computational resources and ensure complete anonymity, this study focuses on the lightweight security of PKC-trusted entities. Consider the use-case OSMARS, supported by a lightweight probabilistic cryptographic scheme, called MBLT. The cryptographic architecture for OSMARS comprises two tiers of built-in administrations. The first tier involves assigned administrators **(A_{a1}, A_{a2}, ., ., ., A_{an})** entrusted with guiding online customers through payment processes using payment cards, as illustrated in Figure 1. The second tier involves the super-administrator (*Sa*) on the e-transactional cloud, who assigns smart level-1 administrators restricted privileges. These administrators are primarily responsible for coordinating and assisting registered customers in the OSMARS-ECTS system.

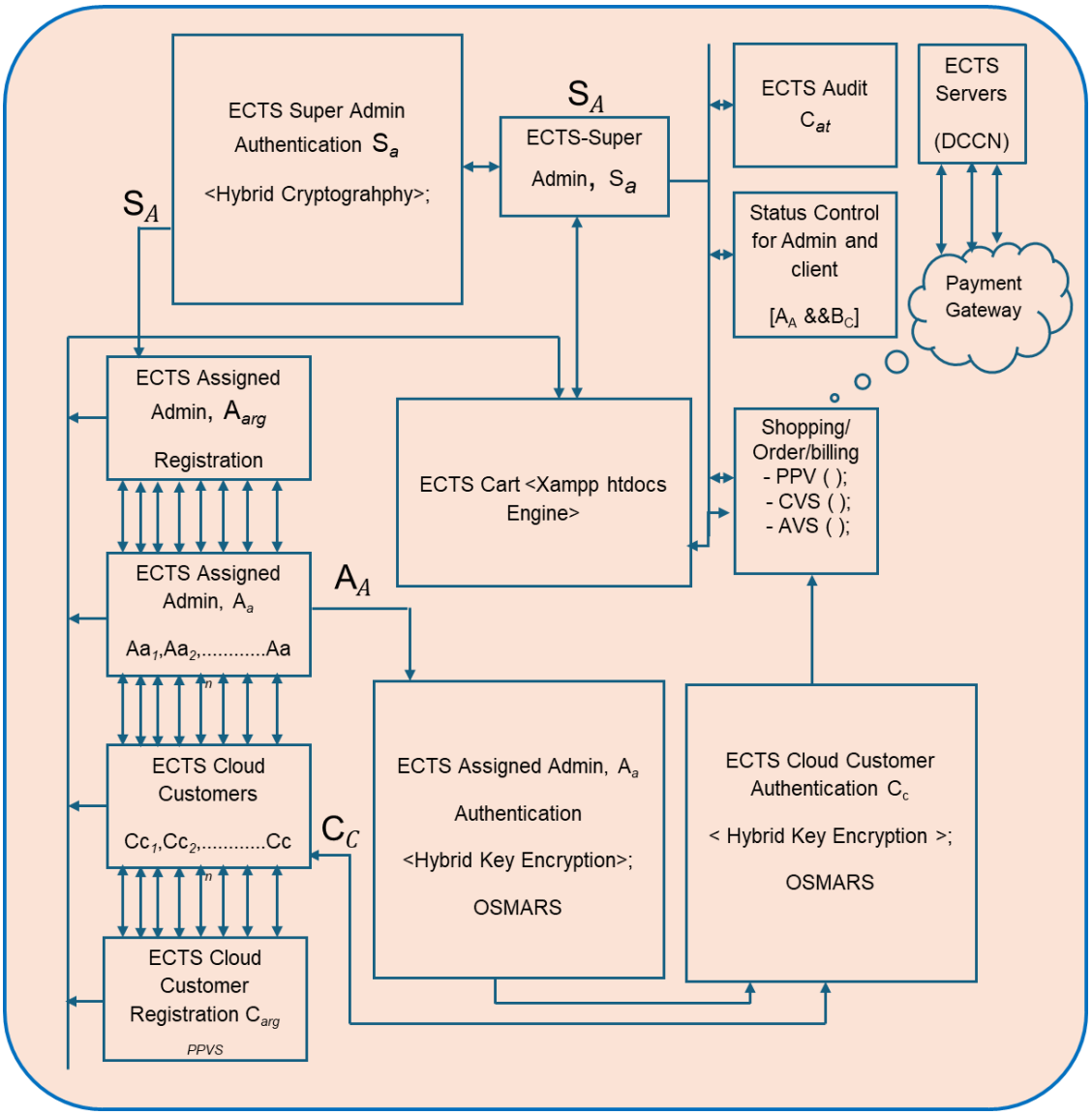

Figure 1. Proposed Computational Front-End Architecture for SaaS ECTS.

- ***Customer Journey and Authentication Mechanisms in OSMARS***

As depicted in Figure 1, Customers of ECTS cloud ***(Cc1, Cc2, ..., Ccn)*** are seamlessly guarded throughout their online journey until the payment stage is successfully completed via the payment gateway. Customer registration involves a dual-factor authentication approach that combines probability distribution models with hybrid cryptographic schemes. A novel concept, Naive Bayesian model is introduced in this paper and elaborated upon in subsequent sections. Lightweight access control, authentication, and encryption mechanisms are employed to grant or restrict access to the OSMARS domain based on the operational status of the assigned administrators (A_a). The status control feature enforces either the ***Aa*** or ***Cc*** roles, whereas the cloud audit function securely stores cloud logs for both ***Aa*** and ***Cc*** activities.

- ***SaaS Reference Architecture***

In this section, we describe the architecture of the SaaS e-payment system, focusing on a use case where an online customer securely processes a payment using a credit card through a web-based checkout or chip reader.
The system involves four core entities: the Client-facing Registration Interface (CFRI), Token Hard Phase (THP), Payment Card Transaction Switching and Processing Institution, and the Cloud Service Provider. Each entity plays a crucial role in ensuring secure, reliable, and efficient transaction management. As illustrated in Figure 1, the process starts when a client enters card information online. This initiates a structured e-transactional procedure that involves the cardholder, vendor merchant, and immediate host intermediary, facilitating a secure and efficient data exchange.

Additionally, the architecture features a second layer that includes administrative and high-level super-administrative roles to enhance manageability and control mechanisms. We highlight the components of the SaaS e-payment system and transactional flow, emphasizing the role of each entity in ensuring secure and efficient electronic transaction. The figure also shows that the joint likelihood function connects clients/customers with both payment gateway and cloud dependencies. In Section 4, we demonstrate that our algorithms support lightweight encryption computations, enabling complex processes that occur billions of times daily, thereby improving the fragile and vulnerable payment ecosystem.

- ***Optimizing Customer Experience in Online Smart Malls***

Figure 1 illustrates an e-payment scenario using a digital card for service payment, detailing processes like subscription profiling, debit card usage, and order review. Key transaction details such as the order number, date, shipping address, and payment specifics are included. The e-transaction structure integrates five entities into a cohesive system optimized by the OSMARS cryptographic architecture, with each entity playing a distinct role in the payment ecosystem. Robust security mechanisms ensure data protection, efficiency, and compliance, enabling seamless and secure transaction flows while addressing vulnerabilities in traditional payment methods.
The design, as depicted in Figure 1, shows how each entity operates in synergy to provide a secure and efficient e-payment system, with the OSMARS cryptographic architecture optimizing every stage of the transaction. Below explains how the five key entities fit into the design illustrated in Figure 1.

1. Initiator (The Buyer): The initiator represents the registered customer interacting with the system through the Client-facing Registration Interface (CFRI). The process begins when the buyer inputs payment details, triggering the OSMARS cryptographic architecture to authenticate their identity using hybrid cryptographic schemes and Naive Bayesian models. The buyer's actions, such as order creation and payment initiation, are securely processed and logged. Administrators *(Aa)* ensure secure data handling and guide the initiator through the transaction.
2. Merchant (Service Vendor): The merchant is depicted in Figure 1 as the entity operating the payment gateway, which directly interfaces with the initiator. They use the Token Hard Phase (THP) to tokenize sensitive customer data, such as credit card information, ensuring secure transmission to downstream entities. Merchants comply with transaction protocols monitored by the Clearing-House Network (CHN), which guarantees accountability and traceability.
3. Merchant Acquirer/Financial Institutions: The merchant acquirer, positioned as a key intermediary, connects merchants to the issuer bank. In Figure 1, acquirers interact with the Payment Card Transaction Switching and Processing Institution. The OSMARS framework ensures acquirers securely handle high transaction volumes with lightweight encryption and cloud auditing mechanisms to log transaction activities and manage risks.
4. Issuer Bank: This is responsible for managing the customer's card account, appears in Figure 1 as the entity validating and authorizing payments. Communication between the issuer bank and the CHN is encrypted to maintain security. Administrators (Aa) and super-administrators (Sa) monitor payment authorizations, leveraging the OSMARS cryptographic architecture to prevent fraud and minimize processing errors.
5. Clearing-House Network (CHN): Central to the design, the CHN ensures seamless data flow and compliance between all entities. In Figure 1, the CHN acts as the backbone, linking customers (initiators), merchants, financial institutions, and issuer banks while enforcing rules, managing fees, and resolving disputes. The joint likelihood function depicted in the design represents the role of the CHN in coordinating transaction approval processes, supported by lightweight cryptographic algorithms to maintain speed and efficiency.

- ***Transaction Stages and ECTS Features***

Two-Tier administration framework were applied throughout the various phases of the ECTS. The system assigns first-tier administrators (Aa) to assist customers and merchants, ensuring a smooth journey through the OSMARS-ECTS system. Super-administrators (Sa) oversee operations at a higher level, managing privileges to maintain system integrity such as auditing and transaction status checks. The key stages of a successful transaction include validation using asymmetric cryptography to verify client account balances. Merchant batching, clearing, and funding processes may introduce delays before the issuing bank is paid. The Electronic Card Transaction System (ECTS) supports diverse e-payment methods such as credit and debit cards, smart cards, e-money, fund transfers, mobile wallets, virtual cards, and tokenization. Popular mobile payment apps, including Alipay, WeChat Pay, Apple Pay, Samsung Pay, and Google Pay, address some traditional e-payment limitations but still face challenges. Within the proposed ECTS, the three key authentication schemes (shown in Figure 1) are as follows:

i. **Password-Protected Verification Software (PPVS):** Similar to how customers access banking portals using passwords, many card networks use PPVS with unique passwords to authenticate cardholders. The system requires cardholders to register and create a unique password ID for transactions, reducing fraud risk.

ii. **Card Verification Value (CVV):** A three or four-digit code distinct from the card chip strip that identifies the cardholder. Merchants use this code to authenticate the cardholder.
iii. **Address Verification Service (AVS):** Requires cardholders to provide address details matching their card issuer's profile. For online payments, merchants request AVS details linked to the billing address, street number, and ZIP code. A mismatch leads to a decline in the number of transactions.

3.4. Modeling Transaction Dynamics with HPPM

In this subsection, we derive the Homogeneous Poisson Process Model (HPPM). The idea in this Poisson probability model is to estimate the expected count of key-encrypted transactions per customer at any given instance. This model facilitates calculation of the prior probability of customers within the ECTS system. Subsequently, Bayesian Model (BM) is invoked to compute the posterior probability of each customer, building upon their respective priors. Again, we establish a uniform minimum variance unbiased estimator (UNVUE). The essence is to ensure an equitable access time, that is a consistent mean rate of server access for all customers utilizing their random PINs, regardless of their geographical locations. This implies that it offers a reliable estimation of the timestamp without bias while reducing errors. In this work computational data for SaaS e-commerce transactions will be collected through an optimized simulation process, ensuring the alignment of the model with real-world scenarios.

Let MBLT $(G) = (V, E)$ be a digraph
let $c: E \to R_0^+$ be a mapping; $c(e) =$ edge e capacity, let s and t be two special vertices of MBLT(G) such that t is accessible from s[1].

Hence $N = (G, c, s, t)$ is called a Bayesian flow network with adversarial source s and sink t.

The probabilistic flow on N is a mapping $f: E \to R_0^+$ satisfying these MBLT(G) key conditions (Equations (5) and (6):

(F_1) $0 \leq f(e) \leq c(e)$ for each edge e; (5)

(F_2) $\sum_{e^+=v} f(e) = \sum_{e^-=v} f(e)$ for each vertex $v \neq s, t$, (6)

where e^- and e^+ depicts the beginning and end of vertex e, respectively. The feasibility condition (F_1) requires each customer's transaction to be encrypted with nonnegative flow traffic. It should not exceed the edge capacity e. The flow encryption condition F_2 refers to the preserved message flows at each vertex, from the source to the sink. Overall, the encryption flow at time t has prior and posterior probabilities, as discussed in Section 4.

3.5. Memory-Based Lightweight Tokenization (MBLT)

In this study, MBLT secures access to resources in e-platform services, particularly during e-transaction processes, by addressing potential attack vectors related to authentication tokens. The MBLT updates transaction probabilities using posterior probability (PP) based on recent information and employs Bayesian principles to calculate PP, leveraging prior probability. It adjusts the prior probability to PP through a distribution based on prior data and integrates it with an access authentication model to block unauthorized access and enhance e-commerce security. Characterized by its concise and lightweight framework, MBLT uses a probabilistic approach for authentication, and adaptively updates access probabilities through memory-based tokenization to effectively prevent unauthorized access. In the subsequent section, specifically Section 3.5.1, we present the access authentication model integrated with the MBLT. This model was devised to eradicate unauthorized access and fortify vulnerabilities within the e-commerce system. The characterization of the MBLT for access authentication, incorporating both prior and posterior probabilities, is discussed using our lightweight framework in Subsection 3.5.1.

3.5.1. Lightweight Characterization

Considering Figure 2, let *M* represent the first gathering customer *i……. n+1* in the MBLT. The Client $\boldsymbol{A_i}$ can supply information such as Names, PINs, Signatures, Biometric data, using any IoT edge device. Each piece of information from $\boldsymbol{A_i}$ is from probability distribution space describing each client while using asymmetric public keys (LKEM). The probability distributions of each client $\boldsymbol{A_{i+1}}$ are collectively hosted by the likelihood function. This represents the collection of all the probability distribution functions of clients $\boldsymbol{A_{i+1}}$. For security purposes, the data supplied by $\boldsymbol{A_{i+1}}$ are obtained using public keys. This key is encrypted strongly with lightweight asymmetric keys (private and public keys). For any transactional access to e-commerce servers, lightweight private keys are used during encryption while public keys are used during decryption from the cloud server. The key attributes of the MBLT adopted in this study are briefly explained below as applied in Algorithms 1 and 2.

- **Probability distribution** ($\prod_{k=1}^{n} P_d$)**:** This is a model function derived from operational probability theory. This is used to estimate the likelihood that various possible outcomes of an ECTS experiment will occur (customers, attackers, transactions., etc). In terms of its sample space and event probability, it is a mathematical description of a random phenomenon in subsets of the sample space (ECTS).
- **Prior probability distribution** $\sum_{\substack{0 \le i \le m \\ 0<j<n}} P(i,j)$**:** In Bayesian statistical inference, the prior is the probability distribution that describes an assumption about an uncertain quantity (user prior key) before certain data (posterior) key) is considered.
- **Posterior probability distribution** $\cup_{n=1}^{m}(X_n \cap Y_n)$**:** This is the probability distribution of an unknown quantity (user prior key) handled as a random variable and conditional on the results of an algorithm.
- **Discrete probability distribution** $\sum_{\substack{0 \le i \le m \\ 0<j<n}} P(i,j)$: This is the probability mass function relevant in situations where the collection of possible customers is discrete, and the probabilities are represented by a discrete list of represented probabilities. ***(i, m, j, n)*** are distribution variables.
- **Poisson distribution** $\sum_k \binom{n}{k}$: This is the discrete probability distribution used in applied probability theory to depict the likelihood that a given event sample (e.g., customer registration, transactions, etc) will occur within a specified time space at a known constant mean rate ***(n).***

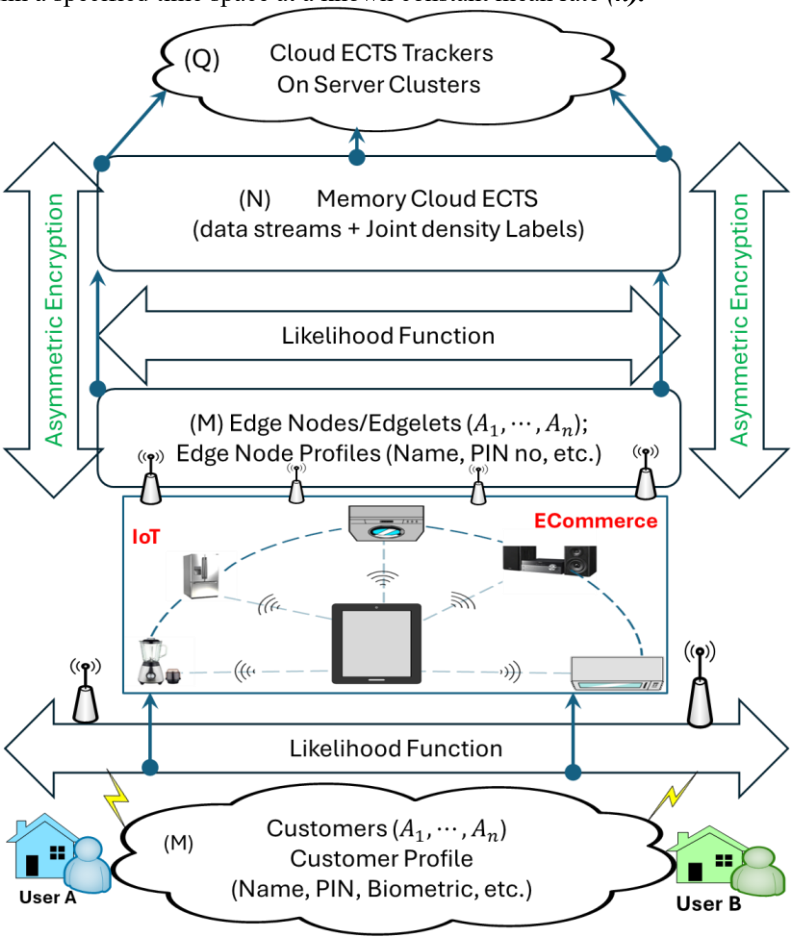

Figure 2. Proposed ECTS for HKCE-MBLT implementation

The public keys are encrypted in such a way that only a specific A_{i+1} knows the data transactions at any instance *t*. Then, the message data in ***M*** is collected, processed, and stored in ***N***. Each of the data is stored in a separate memory. The MBLT processes and the captured data are stored in server memory. If the supplied data at ***M*** matches the data stored in server memory, access is guaranteed, and users can complete transactions quickly. The processed data in ***N*** is sent to ***Q*** for storage and dispersion on demand. ***Q*** has a wider coverage area and can provide an equal platform for the data to be accessed by A_{i+1} irrespective of their locations.

To model this MBLT problem scenario, it's assumed that each customer A_i ***(i = 1, 2, . . ., n or*** A_{i+1}***)*** is an independent random variable that come to the service cloud server for on-demand e-commerce transactions. Hence, , A_i has a Poisson distribution for each customer. Each A_i executes a given number of transactions (that is., $tr_1, tr_2, tr_3 \ldots tr_{n+1}$**)** at a given length of time, T_n**.** This implies that the transaction is a function of time.

Algorithm 1. MBLT Computational algorithm for the Prior and Posterior probability distributions

1. Inputs: $A_i = x_i; i = 1, 2, \ldots \ldots \ldots, n$ where $A_i = x_i$ are random variables, and Call_Schedule (MBLT)

2. Begin ( ) Joint distribution (*jd*) ; *m* signatures

int *i*←0;
for $A_i = i \ \ldots\ldots n+1$
Call $[P(A_i = a_i) = P(X_i = x_i)]$
/* the probability that the random variable A_i assumes a value a_i is equal to the probability that the random variable X_i assumes value x_i */
if $P(x_i) = P(x_2) \ldots\ldots P(X_n)$ ***Then***
/* each of the random variables above has an equal probability of occurrence */
$P(x_1 = x_i) = \frac{\theta^x \exp(-\theta)}{x!}; x = 0,1,2, \Lambda, \infty$ (*Pmf* of Poisson distribution)
/*Because the random variables follow the Poisson distribution, the probability mass function (*pmf*) is computed as*/.
pmf ← random variables ()

$$P(x_1; \theta) = \sum_{x=0}^{\infty} \frac{(\theta)^x \exp(-\theta)}{x!}$$

end if
/*For many of the random variables (i.e., many customers) */
/*But since each of the random variables transacts business with the server at a given time, that is, $P(x_1')$ is associated with time*/,
if $\theta = \theta t$,
$P(x_1; \theta_2) = P(x_2; \theta_2) = P(x_n; \theta_n)$ ***Then***

$$P(x_1; \theta t) = \sum_{x=0}^{\infty} \frac{(\theta t)^x \exp(-\theta t)}{x!}$$

/*map→ Non-homogenous Poisson process with parameters θt */.
/*Likelihood (joint distribution) function of the distribution is computed below*/.
$P(x_n; \theta t) = \prod_{x=1}^{n} \frac{(\theta t)^x \exp(-\theta t)}{x!}$ /* joint distribution */
$P(x_n; \theta t) = \frac{(\theta t)^{\sum x_i}}{\prod x!} \exp(-\theta t)$ /* joint distribution */
end if
end for
If $\prod_{x=1}^{n} x! = N!$ ***Then***
Sterling's formula is valid:
$N! = \sqrt{2\pi}.N^{N+\frac{1}{2}}.\exp(-N)$ *where* $S = N! = n!$
/*Since *N!* is large, the approximate value of *N!* is given by Sterling's formula*/
$\log_e S = \frac{1}{2} log_e 2\pi + \left(N + \frac{1}{2}\right) log_e N - N$
$S = \exp\left(\frac{1}{2} log_e 2\pi + \left(N + \frac{1}{2}\right) log_e N - N\right) \leftarrow Trans(\log_e S)$
endif
Call (power law process (PLP)).
Where *i* < ECTS_ Call_Schedule (MBLT) *does*
historyItem←HistoryList.get (Encryption.List.*Size*())
ECTS_weight ← *Joint density function*;
LightweightedMoving←*LightweightedMoving*+ (*ContainerhistoryItem* * *keylength*);
total Lightweight ←*total light_weight* + *Joint_density_weight*;
i ++;
end while
MBLT ← *LightweightedMoving (m signature are valid)* /
Return *A_DES (decryption)*

Hence, the intensity (mean) of this Poisson distribution is associated with the time $\boldsymbol{T_n}$, therefore $\boldsymbol{A_i}$ **~NHPP (** θt **).** However, the collective information given by A_i needs to be aggregated and stored in cloud memory. Hence, the joint density (likelihood) function collects this information (***M***) as shown in Figure 2. This is then sent it to (***N***)

Algorithm 2. MBLT with Power Law Process (PLP)

Input $A_i = x_i; i = 1, 2, \ldots\ldots\ldots, n$ where $A_i = x_i$ are random variables; Call_Schedule (MBLT) intensity function_monitorCallSchedule ( ) ; posterior probability ( ); sufficient statistic *Status* //
A_i_*decision*←Empty;
PLP←Empty;

// MBLT_Monitor

recycle every *monitorCallPeriod* minutes

// Power Law Process//

If $\Lambda(t) = \theta t^b$ **then**

/**call* intensity function of PLP */

$\frac{d}{dt}\Lambda(t) = \theta b t^{b-1}$

If, $\theta = a$ ***then***

$\frac{d}{dt}\Lambda(t) = abt^{b-1}$

/* Where ***b*** = the shape parameter, ***a*** = the intercept on-axis.

If b = 1, then

HPP(θ); /*implying a stationery increment/

if b >1, or b < 1, then

NHPP (θt) /*with either increasing or decreasing intensity. */

To determine a and b

$log_e\Lambda(t) = log_e a + b log_e t$ /* b = slope; a = intercept on $log_e\Lambda(t)$*/

/* A plot $log_e\Lambda(t)$ against $log_e a$ will give the value of b, the slope */

If *jdf*= True, ***then***

$P(x_n; \theta)$ = True /*the probability of several transactions per customer at a given time t*/

$P(x_n; \theta) = \frac{(abt^{b-1})^{\sum x_i}}{N!} = \exp(-abt^{b-1})\{\frac{(abt^{b-1})^{\sum_{i=1}^{n} x_i}}{S}\}$

/*The probability model for the e-commerce transaction*/

$P_n(t) = \exp\{-abt^{b-1}\}[\{-abt^{b-1}\}^n]/n!$

/* To determine the posterior probability, then apply the Bayesian Statistics model) */

$$P\left(\frac{y_{A_i}}{\theta_{A_i}}\right) = \frac{P(\theta_{A_i}).P(^{y_{A_i}}/_{\theta_{A_i}})}{\sum_{i=1}^{n} P(\theta_{A_i}).P(^{y_{A_i}}/_{\theta_{A_i}})} = \frac{P(abt^{b-1}.A_i).P(^{y_{A_i}}/_{abt^{b-1}.A_i})}{\sum_{i=1}^{n} P(abt^{b-1}.A_i).P(^{y_{A_i}}/_{abt^{b-1}.A_i})}$$

/* $P\left(\frac{y_{A_i}}{\theta_{A_i}}\right)$ probability of obtaining information about A_i*/

History (all the information) A_i ← Valid

PIN identifier ← Valid /*Cryptographic attribute needed in the design*/

if scalingParameterType is a random number (PIN code) Aware ***then***

Jdf_Status←Predictor (*history* of $x_i = r_{Ai}, r_{Ai} \subset A_i$ *or* $r_{Ai} = A_i$)

/* stored in ECTS server alongside the A_i information in the memory */

elseif *scalingParameterType* is NotPIN-Aware ***then***

No Jdf←Predictor ($\boldsymbol{A_{i+n}}$)

end if

/*Probability that the information stored about A_i's in *(M)* corresponds to the one in the server memory *(N)*/

If *Jdf* is True, ***then***

decision←Scale Up

$P(r_{Ai}) = P(^{r_{Ai}}/_{y_{Ai}})$

Elseif *parameterStatus* is True ***then***

$$P(^{r_{Ai}}/_{y_{Ai}}) = \frac{P(r_{Ai}).P(^{y_{Ai}}/_{r_{Ai}})}{\sum_{i=1}^{n} P(r_{Ai}).P(^{y_{Ai}}/_{r_{Ai}})} = P(^{y_{Ai}}/_{\theta_{Ai}})$$

sufficient statistics ←$T_{n(x)} = \frac{y(y-1)}{n^2}$

end if

// MDLT Executer

If $P(^{r_{Ai}}/_{y_{Ai}})$ is True ***then***

PIN (stored about A_i's in *(M)* = Server memory *(N),*;

Elseif *decision* is True ***then***

Call UMVUE /*uniform minimum variance unbiased estimator *( )*/;
prior probability distribution *( )*

end if

end if

end if

end if

end return

where it is stored as raw data, processed, and sent to the server memory. The distribution function that can remember (store and recall) the history of A_i and relate it with the current information about A_i is Bayesian Statistic.

Now, there is a need for posterior probability. This provides current chances of obtaining transaction information about the customer, given that the information has been collected and stored previously via their digital card. This is because the customer can access the e-commerce server certificate authority using strong encryption. Consequently, we developed a Bayesian construct for this scenario as illustrated in Figure 1. In the proposed MBLT, the Bayesian statistic has memory and can link prior information about A_i to the current information about A_i and store them. The legacy e-commerce information is stored with an elliptical curve cryptosystem, Paillier, and ElGamal Encryption subject to the attack model. The e-commerce server then activates the authentication process. However, server access from the cloud is linked to the initial (prior) message given by the A_i; if it corresponds to it, then A_i can be served by the server. Hence, the posterior probability of A_i links to the prior probability of A_i to generate a result (i.e, for A_i to be served). The information supplied by A_i, such as PIN, must be encrypted with the public key for security purposes and on accessing the server by the customer, which will be decrypted by A_i (the customer for security purposes) only; where this information leaks, A_i will be vulnerable to fraud, cyber attacks and cyber insecurity.

To avoid such a vulnerability attack vector, Algorithm 2 computes a random number that uniquely identifies every customer for transaction security. This ID is encrypted using lightweight public keys but can only be decrypted by the customer for secured service access from the server. Random number r_i = PIN = subset of the information supplied by A_i. This is generated such that the probability that r_i = PIN, given that the prior information about A_i is known. Now, we develop the probability model for e-commerce authentication and determine an unbiased estimator for uniform e-commerce access regardless of customers' location. The formulated model was derived as follows.

From Table 3, the slope is determined using Algorithm 2 as $b = 1$. From the relationship in Algorithm 2 (where $log_e \Lambda(t) = log_e a + b log_e t$), a new computation is derived *such* that $\log_e(a) = 1.7$, calculated as $a = \exp(1.7) = 5.4739$. Hence, the prior probability distribution is derived in Equation (7), whereas Equation (8) represents the posterior probability.

$$P_n(t) = \exp\{-a\}\left[\{a\}^{\Sigma x}\right]/S = \exp\{-5.4739\}[\{5.4739\}^n]/n_! \quad (7)$$

$$P\left(\frac{y_{A_i}}{\theta_{A_i}}\right) = \frac{P(\theta_{A_i}).P(^{y_{A_i}}/_{\theta_{A_i}})}{\sum_{i=1}^{n} P(\theta_{A_i}).P(^{y_{A_i}}/_{\theta_{A_i}})} = \frac{P(abt^{b-1}.A_i).P(^{y_{A_i}}/_{abt^{b-1}.A_i})}{\sum_{i=1}^{n} P(abt^{b-1}.A_i).P(^{y_{A_i}}/_{abt^{b-1}.A_i})} \quad (8)$$

Where

- b: Slope of the logarithmic relationship, which is determined to be 1 in this case from Algorithm 2.
- a: A constant derived from the relationship
- $P_n(t)$: Prior probability distribution at time T, as shown in Equation (7). This is the probability distribution before the new evidence is introduced.
- x: represents a variable in the summation of a transaction event count in the context of probability distributions
- n: Number of events, representing the number of transactions/ observations.
- S: A normalization factor or the total sum of probabilities, ensuring that the prior distribution is valid and sums to one.
- $P\left(\frac{y_{A_i}}{\theta_{A_i}}\right)$: The posterior probability of an event y related to the customer y_{A_i}, given parameter θ_{A_i}, as shown in Equation (8). This is the probability of being updated with the new evidence data.
- θ_{A_i}: Parameter representing prior knowledge or characteristics related to customer A_i.
- abt^{b-1}: Term in the posterior probability formula representing the intensity of the process, derived from the Poisson model. It captures the rate of events occurring over time, t.
- $\sum_{i=1}^{n} P(\theta_{A_i}).P(^{y_{A_i}}/_{\theta_{A_i}})$: The summation of all customers or events from 1 to n, accounting for the total contribution to the posterior probability.

Thus, Equation (8) computes prior probabilities using exponential distribution. It updates them to posterior probabilities using Bayes' theorem [45], capturing parameters that show the transaction rate and event intensity over period t.

3.5.2. ECT Cryptographic Derivation

In this section, we characterize the lightweight cryptographic structure of a secured ECTS using a low-complexity probability distribution. The derivation of the computational MBLT was achieved with Algorithms 1 and 2 for prior and posterior keys to secure access to ECTs. In addition, Algorithm 2 determines an unbiased uniform estimator for service accessibility in ECTS processes. Table 3 lists the harmonized cryptographic derivation datasets for the digital signature based on Algorithms 1 and 2. The authentication construct needed for the MBLT was built using Minitab and MATLAB [46, 47]. In this study, the customers transacting a given number of e-commerce activities ***(n)*** via the IoT token into the cloud at any given time ***(t)*** are completed in the lightweighted mode. Using the developed algorithms, the average number of transactions made by each customer at a given time is obtained. The parametric computations are presented in Table 2. The slope of the log plot was used to compute the prior probability distribution. This is referred to as the Bayesian statistical inference described previously. On the other hand, the probability distribution of an unknown random variable that is conditional on the results of a prior passphrase is a posterior probability distribution. The idea is that once there is new passphrase information, the previously stored passphrase is immediately updated. Table 4 describes the derived prior, and posterior probability functions used to explain the MBLT digital signatures (DS). Hence, the PKC the digital authentication is governed by Algorithms 1 and 2. The key-scattering techniques [48], and [49] are compared with the proposed scheme to validate this work in Section 6.

Table 3. Parametric computations [n-Tras, t(hrs) ln(t) and $ln(\omega t)$for $\omega = 5; t = 4$]

Cstm	n-Tras	t(Sec)	ωt	$ln(t)$	$ln(\omega t)$
1	8	5	25	1.609438	3.218876
2	6	5	25	1.609438	3.218876
3	9	3	15	1.098612	2.708050
4	7	1	5	0.000000	1.609438
5	9	5	25	1.609438	3.218876
6	7	2	10	0.693147	2.302585
7	7	3	15	1.098612	2.70805
8	4	1	5	0.000000	1.609438
9	7	3	15	1.098612	2.708050
10	5	5	25	1.609438	3.218876

Table 4. MBLT No. of Transactions, Pn(t), Joint Probabilities, Prior and Posterior probabilities (DS)

Cstm	n-Trans	A	Exp(-a)	a^n	Prob.	Prop. N	Prior Pr.	Joint pr.	Poste. Pr
1	8	5.4739	0.004195	806073.9	0.083863	0.068376	0.005734	0.122948	0.046639
2	6	5.4739	0.004195	26901.79	0.156734	0.051282	0.008038	0.122948	0.065374
3	9	5.4739	0.004195	4412368	0.051006	0.076923	0.003924	0.122948	0.031912
4	7	5.4739	0.004195	147257.7	0.122564	0.059829	0.007333	0.122948	0.059642
5	9	5.4739	0.004195	4412368	0.051006	0.076923	0.003924	0.122948	0.031912
6	7	5.4739	0.004195	147257.7	0.122564	0.059829	0.007333	0.122948	0.059642
7	7	5.4739	0.004195	147257.7	0.122564	0.059829	0.007333	0.122948	0.059642
8	4	5.4739	0.004195	897.8162	0.156925	0.034188	0.005365	0.122948	0.043636
9	7	5.4739	0.004195	147257.7	0.122564	0.059829	0.007333	0.122948	0.059642
10	5	5.4739	0.004195	4914.556	0.171798	0.042735	0.007342	0.122948	0.059715

From Equation (9), the computation for the UMVUE is now completed in MBLT. This UMVUE refers to the average time each customer will spend accessing SaaS e-commerce services as depicted in Figure 2. This is computed as follows. From Table 1,

$$T_{n(x)} = \frac{y(y-1)}{n^2}; where\ y = \sum_{i=1}^{n} x_i \quad (9)$$

$$T_{n(x)} = \frac{20(20-1)}{20^2} = 0.95\ secs$$

The uniform average time it will take each customer to access the SaaS server pool and the payment gateway for cryptographic authentication is 0.95 seconds. This is the average time it takes the server to service customers, irrespective of their location. The probability that the information collected and stored by ***M*** on customer "1" is equal to that stored by ***N*** on the same customer is 0.046639 (digital signature). In other words, the probability that the authentication secret PIN of customer "2" is accessed in ***N*** given that the same information was in ***M*** is 0.065374, etc. Section 5 focuses on PKC provisioning on MBLT. Recall that the prior and posterior probability IDs represent the PKI signatures. This is the verifiable key identifier used to verify access in ECTS. It uses a digital certificate and its associated signature. From Table 3, the Naive Bayes classification scheme is predicated on the notion that predictors (prior and posterior probabilities) are independent and can eliminate the attacker's efforts. The proposed MBLT PKE scheme is discussed in Section 5.

4. SaaS Decomposition Design (MBLT PKE)

In this subsection, we detail the protection of the PKE scheme and MBLT posterior signature, as shown in Table 3. We describe how the random computational scheme validates encryption and signature mechanisms. In the SaaS ECTS, we also explore the formalization of MBLT hybrid encryption for implementing digital signatures. Using the OSMARS use case from Figure 1, we present an enhanced asymmetric encryption approach that employs 512 bits in the PKES, and signature algorithms (SA) illustrated in Figure 2. The following components in the customer layer are discussed.

4.5. MBLT Public Key Encryption Against Attacker Vectors

The cryptographic schemes for IoT tokens used in ECTs are discussed in this subsection. Goldwasser–Goldwasser–Micali [50], and Elgamal, and Paillier encryption schemes [51] are required for MBLT hybrid encryption. This provides secure, efficient, and scalable transactions in IoT-driven ECT ecosystems.

4.5.1. Goldwasser–Micali Encryption (GME)

The GME cryptosystem as an asymmetric PKE scheme is a legacy probabilistic encryption scheme against attackers. It offers passive security but can easily be scaled. Now, the MLTS-GME scheme using ECTS hardness can be regarded as a hybrid integer N and an integer e.

In the ECTS, let the set of domain squares be given in $(\mathbb{Z}/N\mathbb{Z})$ be established as Equation (10)

$$Q_N = \{x^2\ (mod) N : x \in\ (\mathbb{Z}/N\mathbb{Z})^*\} \quad (10)$$

and J_N represents the elemental set having Jacobi symbol =1, as given in Equation (11).

$$J_N = \left\{a \in \left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^* : \left(\frac{a}{N}\right) = 1\right\} \quad (11)$$

The entire ECTS pseudo-square yields the difference $J_N \setminus Q_N$. Considering the RSA-representative modulus $N = p * q$, the elemental variables in J_N are $(p-1)*(q-1)/2$. Similarly, the number in Q_N is $(p-1)*(q-1)/4$. The ECTS problem is that given an element x of J_N, it is difficult to tell whether $x \in\ Q_N$, while it is easy to determine whether $x \in\ J_N$ or not. In this context, we discuss the GME cryptosystem below.

Key Generation: As a private key, let's take two large prime numbers $\gamma t = (p, q)$ and then compute the public modulus $N \longleftarrow p.q$, and an integer $y \in J_N \setminus Q_N$. The lightweight asymmetric key is set to be $\beta t = (p, q)$, (i.e., public key). The power of y is the public key owner with the computing elements. $y_p \in F_p^*$ and $y_q \in F_q^*$ such that Equation (12) holds.

$$\left(\frac{y_p}{p}\right) = \left(\frac{y_q}{q}\right) = -1 \quad (12)$$

Thus, the estimation of y value is derived from y_p and y_q based on the remainder Theorem. The y value is generated in such a way as not to depend on in Q_N, but on the map in J_N because of Equation (13)

$$\left(\frac{y}{N}\right) = \left(\frac{y}{p}\right).\left(\frac{y}{q}\right) = \left(\frac{y_p}{p}\right) = \left(\frac{y_q}{q}\right) = (-1).(-1) = 1 \quad (13)$$

Encryption: The GME system has one-bit information at every instant t. Its Bit b encryption is then given by the map below.

- $\boldsymbol{x} \leftarrow (\mathbb{Z}/\boldsymbol{N}\mathbb{Z})^*$
- $\boldsymbol{c} \leftarrow (\mathbb{Z}/\boldsymbol{N}\mathbb{Z})^*$

$\boldsymbol{c}$ is ciphertext. This is not sufficient, as a single bit of plaintext is needed. $\log_2 N$ Ciphertext bits for transmission.

Decryption: It is obvious that J_N is the owner of ciphertext **c** in MBLT hybrid encryption. The value of $\boldsymbol{c}$, is a quadratic residue if the message bit is zero; otherwise, it will be a quadratic nonresidue. Determining whether $\boldsymbol{c}$ is a quadratic residue modulo N requires the decryptor needs to accomplish to retrieve the message. However, it is expected that the decryptor will be aware of N factor, allowing them to calculate the Legendre symbol using Equation (14).

$$Nf = \left(\frac{c}{p}\right) \tag{14}$$

The message bit is zero once this Legendre symbol (LS) = +1 because c is a quadratic residue in that case. The message bit is one, however, if LS = -1, this indicates that c does not represent a quadratic residue. Assuming that the ECTS problem is difficult for RSA style moduli N of size υ bits, it is now quite simple to demonstrate that the Goldwasser-Micali encryption technique is insecure.

Theorem 1.1. Assume that adversary A works against the GME security scheme for moduli of υ size bits, while adversary B works against the ECTS problem. This is given as in the MBLT hybrid encryption, Equation (15).

$$Adv_{\Pi}^{GME}(A) = 2.Adv_{\Pi}^{ETCS}(B) \tag{15}$$

PROOF. An outline for Algorithm B is provided which entails the use of Computational Engine A in MBLT hybrid encryption. Figure 1 depicts this concept. Assume that algorithm B is given N and $j \in J_N$ and is asked to determine whether $j \in Q_N$.
Algorithm B first randomizes j to form y, based on the assumption that j does not lie in Q_N. Thus Algorithm B sets $y \longleftarrow j.z^2\ (mod\ N)$, for some $z \longleftarrow ((\mathbb{Z}/N\mathbb{Z})^*)$. Algorithm A is then given the public key $\beta t \longleftarrow (N, y)$
Because the GME system encrypts bits, it is safe to assume that the find stage of adversary A will simply output the two messages $m_o = 0$ and $m_1 = 1$ and we now form the challenge ciphertext
$c^* \longleftarrow y^b.r^2$

for some $r \longleftarrow (\mathbb{Z}/N\mathbb{Z})^*$ and some bit $b \longleftarrow \{0,1\}$ selected by algorithm B, Algorithm A, will react with its best guess b' for bit if $b = b'$, in which case it will return j as a quadratic residue; otherwise, it will provide the opposite.

When examining the probability, it was discovered that the value of c^* serves as legitimate encryption for message m_b if j is not a quadratic residue. Thus, Algorithm B presents a legitimate rival of Algorithm A if j is a quadratic residue. However, this is not a legitimate encryption of anything if j is not a quadratic residue (because the public key is not even valid). This gives Equation (16)

$$\begin{aligned} Adv_{\Pi}^{ETCS}(B) &= |P_r[\ b' = b | y \in Q_N] - |P_r[\ b' = b | y \in \backslash\ Q_N\ ] \\ &= \left[P_r[A\ wins\ for\ a\ valid\ challenger] - \frac{1}{2}\right] \\ &= \frac{1}{2}.Adv_{\Pi}^{IND-CPA}(A) \end{aligned} \tag{16}$$

Theorem 1.2. The GME scheme is not ECTS secure.

PROOF: Let us say that the target ciphertext is $\boldsymbol{c}^*$, and we want to determine which bit b $\boldsymbol{c}^*$Encrypts. Note that $\boldsymbol{c}^* = y^b.x^2(mod\ N)$. However, utilizing the decrypt engine, the plaintext is still accessible even though the restriction forbids the decrypt computational engine to decrypt. $\boldsymbol{c}^*$ such that $\boldsymbol{c} = \boldsymbol{c}^* * z^2(Mod\ N)$ in MBLT hybrid encryption. The representation $z = \in (\mathbb{Z}/N\mathbb{Z}^*)$ is randomized to a certain value. It is now simpler to understand that $\boldsymbol{c}$ is an encryption of the same bit b. Consequently, asking the engine to decrypt $\boldsymbol{c}$, will result in c∗ decryption for the ECTS.

ElGamal Encryption: Although ineffective, the GME system is passively secure against attackers. A straightforward encryption technique that is both effective and passive is truly needed. The discrete logarithm problem serves as the foundation for the ElGamal encryption technique, which is the most straightforward and effective safe ECTS encryption algorithm. Let us investigate the EIGAMAL EEA finite field of encryption.

Domain Parameters: ElGamal encryption uses public parameters that can be shared by several users, in contrast to the RSA method. The domain parameters are as follows:

- p a "large prime", by which we mean one with approximately 2048 bits, such that $p-1$ is divisible by another "medium prime" q of approximately 256 bits.
- g an element of F_p^* of prime order q

, i.e., $g = r^{(p-1)/q} \ (mod\ p) \neq 1$ for some $r \in \mathcal{F}_p^*$.

A public finite abelian group G of prime order q with generator g is produced using the domain parameters (DP).

ElGamal Generation of Encryption Keys: Immediately after these DP are resolved, both public and private keys can be consummated against attackers. The private key γt is selected as an integer $x \leftarrow [0, \ldots\ldots, q-1]$, whereas the public key is fixed as $\beta t \coloneqq h - g^x (mod\ p)$. As noted, the respective users in the RSA need to generate two large primes to set up their key pair (very costly). However, EIGamal encryption users must obtain randomized numbers while executing exponentiation using modular construct to obtain a key pair.

ElGamal Encryption: It Messages are believed to be Group G components. The following steps must be performed to encrypt a message, $m \in G$.

- $k \leftarrow \{0, \ldots\ldots, q-1\}$
- $c_1 \leftarrow g^x$,
- $c_2 \leftarrow m.h^x$,
- $c \leftarrow (c_1, c_2) \in G * G$

Because each message has a unique short-lived key, k, it follows that encrypting a similar message twice would result in two distinct ciphertexts.

Decryption: A ciphertext $c = (c_1, c_2)$, is decrypted by computing the elements in Equation (17)

$$\frac{c_2}{c_1^x} = \frac{m.h^k}{g^{x.k}} = \frac{m.g^{x.k}}{g^{x.k}} = m \tag{17}$$

By providing two outcomes in the passive security situation, it is feasible to highlight the fundamental security findings about ElGamal encryption against attackers. First, ElGamal is less vulnerable if the Diffie-Hellman problem is difficult, while the second states that ElGamal is ECTS compliant if the Decision Diffie-Hellman problem is difficult. The modifications required are described below.

Theorem 1.3. The Diffie-Hellman problem (DHP) has passive encryption against *A* if an attacker exploits probabilistic security over group *G.*

$$Adv_{\Pi}^{Prob}(A) = Adv_{\Pi}^{DHP}(B)$$

PROOF. Algorithm A takes as input a public key h and a target ciphertext $c^*(c_1, c_2)$ and returns the underlying plaintext. Let us show how algorithm 3 is used to create lightweight encryption to address the DHP. Let us assume B is given $X = g^x$ and $Y = g^y$ while asked to resolve the Diffie-Hellman (DH) challenge, that is, to estimate the payload of $g^{x.y}$; Algorithm 3 arranges an ElGamal public key that relies on the DHP input, that is, put $h \leftarrow X = g^x$. Target ciphertext $c^* \leftarrow (c_1, c_2)$, where $c_1 \leftarrow Y = g^y$ and $c_2 \leftarrow G$ (i.e., group random element). Algorithm 3 describes this scenario. Notwithstanding the encouraging findings regarding the security of ElGamal encryption, other methods are still impervious to adaptively selected ciphertext assaults. ElGamal is trivially malleable, and is primarily a constraint. Lightweight hybrid ciphers against attackers are required in KEM

Algorithm 3. Elgamal public key + Diffie–Hellman Transformation for KEM

Input ciphertext c - History of computational KEM against attackers Encryption_monitorCallSchedule
As input $X = g^x \in G$ and $Y = g^y \in G$

Output: Ciphertext c corresponding to message m

$h \leftarrow X = g^x$
$c_1 \leftarrow Y = g^y$
$c_2 \leftarrow G$
$c^* \leftarrow (c_1, c_2)$,
$m = A(c^x, h)$
Return c_2/m
Call B

end return

5. Security Analysis of the Hybrid Key Cryptography Engine (HKCE)

In this section, we analyse the security of the Hybrid Key Cryptography Engine (HKCE) to meet the outlined objectives, including transaction confidentiality, authentication, identity privacy, and defense against threats like Denial-of-Service (DoS) attacks and impersonation studied in [52]. The HKCE combines the Key Encapsulation Mechanism (KEM) and Data Encapsulation Mechanism (DEM) to form a secure hybrid cipher, where KEM handles public key operations, and DEM encrypts the data. This design ensures efficient encryption without compromising security, as adversaries are challenged through indistinguishability games to differentiate encapsulated keys.

5.1. Designing KEM ECTS Hybrid Encryption Scheme

We derive KEM from the perspective of the encryptor, which accepts a public key βt and produces an encapsulation of that key c, and a symmetric key $k \in K$ for use by the owner of the matching private key. The owner of the private key γt can then use their private key, along with the encapsulation c, to recover the symmetric key k. Consequently, the encapsulation mechanism did not receive any messages. Consequently, this leads to the following three algorithms:

- $(\beta t, \gamma t) \leftarrow KeyGen(K)$
- $(c, k) \leftarrow Encap_{\beta t}(\,)$
- $(k) \leftarrow Decap_{\gamma t}(c)$

In terms of exactness, we need all pairs $(\beta t, \gamma t)$ output by $KeyGen\ (K)$
If $(c, k) \leftarrow Encap_{\beta t}(\,)$ then $(k) \leftarrow Decap_{\gamma t}(c) = k$

The security definition of the indistinguishability of encryptions for PKE algorithms serves as the foundation for the security description of KEMs. However, the ECTS now demands that the key generated by a KEM be identical to a random key for the digital signature. Consequently, the following game defines a security game. A random key, called $k_0 \in K$ is generated by the challenger from the space of the symmetric keys produced by the KEM.

• To create a valid key $k_1 \in K$ and its encapsulation c^*, under public key βt, the challenger uses KEM's Encap function.
• The challenger chooses bit b and sends the values k_b, c^* to the adversary.
• The adversary's objective is to determine whether $b = 0\ or\ 1$.

The ECTS with KEM has a specified benefit as shown in Equation (18).

$$Adv_{\Pi}^{ECTS}(A) = 2.[\Pr(A(\beta t, k_b, c^*) = b) - 1/2] \tag{18}$$

The security described above applies to passive attack cases. To define security against adaptively selected ciphertext assaults, it is necessary to grant adversary access to the decapsulation function. Except for the target encapsulation, this decapsulation method returns the key (or the incorrect encapsulation symbol) for whatever encapsulation c^* the adversary chooses.

5.2. Creating Hybrid Encryption

Our KEM/DEM system is derived by combining a KEM (specified by the KeyGen, $Encap_{\beta t}$, $Decap_{\gamma t}$ and which outputs symmetric keys from space k) and a DEM (described by the algorithms e_k, d_k and with key space k) to create a hybrid cipher or PKE scheme. The key generation process used by the public key scheme is identical to that used in the baseline scheme.

- $(k, c_1) \leftarrow Encap_{\beta t}( )$
- $c_2 \leftarrow e_k(m)$
- $c \leftarrow c_1, c_2$

The recipient upon receiving the pair $c = (c_1, c_2)$, performs the following steps to recover m

- $k \leftarrow Decap_{\gamma t}(c_1)$
- $if\ k \leftarrow \prod c\ return\ \prod c$
- $m \leftarrow d_k(c_2)$
- $Return\ m$

Let us discuss Algorithm 4 while pointing out that in-game G_1 (i.e., ECTS probabilistic payment transactions), if A requests that (c_1, c_2) be decrypted, C can legitimately do so if $c_1 \neq c_1^*$. When the last condition is met, the algorithm 4 returns the decryption of c_2 by using its decryption engine. The payment parameters are encrypted before using the digital signature derived from posterior probability. The O_{LR} Computational queries represent the original load requests in the ECTS. It should be noted that until B aborts because A makes an erroneous query, the target ciphertext c_2^* of B is never assigned to its own decryption cost. Furthermore, computational engine C is only used once, as required by the DEM's one-time security.

Algorithm 4. ECTS Hybrid Encryption Computational Compression

$(c_1, c_2) \leftarrow$ **Input:** KeyGenenerator (K);
Call A having input as public key βt.
Call SHA-512 ( )

Output The plaintext *m* corresponding to ciphertext C*;*

/* $A's\ O_{LR}$ computational queries/

A makes an O_{LR} query with messages m_0, m_1
c *passes* m_0, m_1 to its tagged O_{LR} computational engine to obtain c_2^*
$c_1^* \leftarrow Encap_{\beta t}( )$
$c^* \leftarrow (c_1, c_2)$
Return c^*
$c^* \leftarrow (c_1, c_2)$,
$m = A(c^x, h)$
Return c_2/m
/* $A's\ O_{dk}$ computational Queries*/
A makes an O_{dk} query with ciphertext $c = (c_0, c_1)$
if $c_0 \neq {c_0}^*$ then $(k) \leftarrow Decap_{\gamma t}(c_0), m \leftarrow d_k(c_1)$
else if $c_1 \neq {c_1}^*$ then C passes c_1 to its O_{dk} computation to obtain m
else escape

$$m_2 \leftarrow D(C) = \left(\frac{\left(C^{\beta t} mod\ n^2\right) - 1}{n}\right). \beta t(mod n) * m$$

Return m_2
/* $A's\ response$ */
When $A\ returns\ b'$

end Return

Let us now consider the most difficult part of the process. Algorithm 4 employs KEM to A to compromise ECTS security. The aim is to limit the chance of occurrence through an adversarial response. It carries out the following: The key/encapsulation pair looks out for both genuine encapsulation and fraudulent keys using SHA-512. In the first scenario, A plays game G_0, while in the second, is plays game G_1. This creates an environment for A to play in the adversarial mode.

Algorithm 5. Hybrid Encryption in ECTS Probability Distribution

Input βt, customer's PCs, random numbers e_k, m_b, d_k and timestamp OTP t_k, T_k;
Call A with input the public key βt.
Call SHA-512 ( )

/* $A's\ O_{LR}$ computational queries/

A makes an O_{LR} query with messages m_0, m_1
B *calls* its own O_{LR} computation to obtain c_1^*, k^*
$b \leftarrow \{0, 1\}$

$c_2^* \leftarrow e_{k^*}(m_b)$
$c^* \leftarrow (c_1^*, c_2^*)$
Return c^*
/∗ $A's$ O_{dk} computational queries ∗/
A makes an O_{dk} query with ciphertext $c = (c_0, c_1)$
Return c^*
if $c_0 \neq {c_0}^*$ then
B *calls* its $O_{Decap_{\gamma t}}$ computation on c_0 to obtain k
$m \leftarrow d_k(c_1)$
else if $c_1 \neq {c_1}^*$ then $m \leftarrow d_{k^*}(c_1)$
else escape
return m
/∗ $A's\ response$ ∗/
When $A\ returns\ b'$
$a \leftarrow 0$
$if\ b' \neq b\ then\ a \leftarrow 1$
end return

The algorithm uses the compression function advantage of *A* to differentiate between the two probabilistic Bayes and enforce the KEM. In any ECTS setup, the e-payment subsystem's complexity demands lightweight deployment while maintaining a robust and secure card payment system within an acceptable quality of service metric. Advanced malicious hackers, as shown in Figure 1, can cause losses in credit card networks through fraudulent transactions and data breaches. Without reputation schemes to protect customers from attack tampering and privacy breaches, the ECTS remains vulnerable. To assess the computational efficiency of the proposed scheme, this study uses the PPVF Cheng *et al*. model [77] within the HKCE-MBLT scheme. Clients perform transaction registration steps using the HKCE-MBLT scheme over the network communication channel. Additionally, the interaction between the assigned admin and super admin was secured in the cloud domain. Vulnerable computational elements include the CPU, memory, load balancers, and firewalls. Because asymmetric cryptographic phases occur over the public secure Internet, a certificate authority (CA) is essential for ECTS.

5.3. Platform Design and Security Architecture with PKI

In Figure 2, we introduced the proposed PKI powered by a distributed Joint Density Function (JDF) in Figure 3, supporting the lightweight OSMARS application. This design includes access authentication token APIs to ensure the security of the ECTS system. The JDF elastically provisions system availability and orchestrates secure on-demand services for transactional processes, such as POS, Web Master, and payment terminals. The OSMARS workload is secured during user transactions, utilizing prior and posterior probabilities for authentication. To enhance flexibility and reduce computational constraints, a microservices architecture is implemented, isolating the encryption model and employing event-driven simulation for cryptographic performance evaluation using the work as a baseline [52]. As a result, the computational complexity of the algorithms is discussed in Section 5.4.
Figure 3 illustrates the Joint Density Function (JDF) for implementing the Hybrid Key Cryptographic Engine (HKCE) and Memory-Based Lightweight Tokenization (MBLT)**,** showcasing a secure and efficient cryptographic pipeline. It processes sensitive input data through hybrid encryption (symmetric for bulk data, asymmetric for key exchange) and derives unique cryptographic keys via a dedicated module. The JDF computation enhances randomness and integrates encryption states for optimal performance, bridging the HKCE and MBLT components.
The MBLT transforms processed data into lightweight, secure tokens mapped to the original data using memory-efficient algorithms. The final output is either encrypted or tokenized data, ensuring security, efficiency, and compliance with cryptographic standards as shown in the Algorithm steps below.

- **Step 1**: Input plaintext is fed into the HKCE, where initial encryption operations are performed.
- **Step 2**: Keys derived in the Key Derivation Module enhance security and ensure adaptability for lightweight processing.
- **Step 3:** The processed data and derived keys are fed into the Joint Density Function, which ensures seamless integration between encryption and tokenization processes.
- **Step 4:** The data flows into the MBLT, where lightweight tokenization is performed.
- **Step 5**: The final output is either securely tokenized or encrypted, depending on the use case.

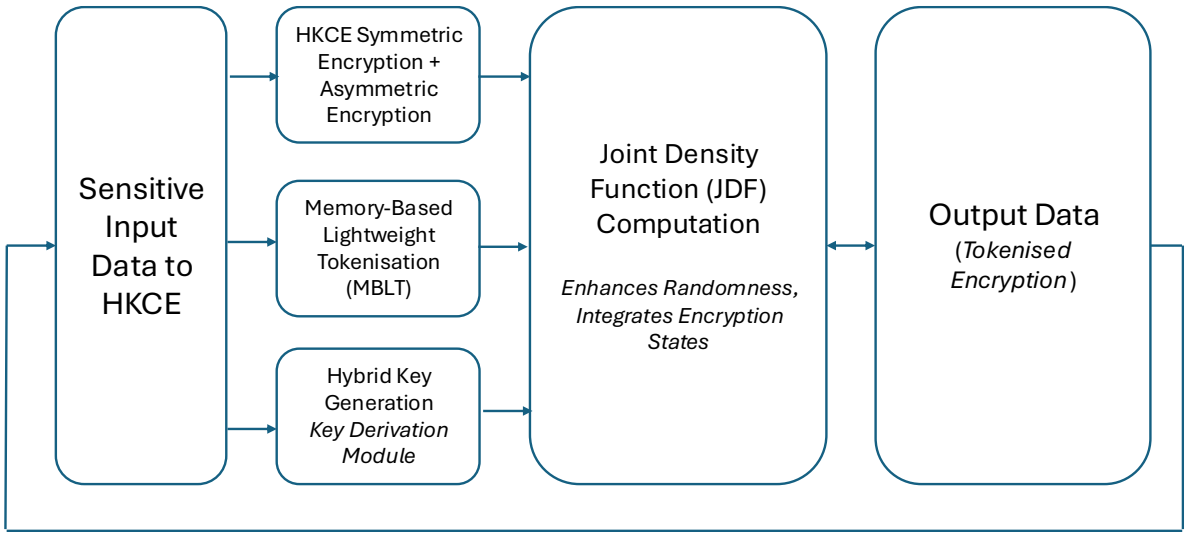

Figure 3: Joint Density Function (JDF) for Secure and Efficient Integration of Hybrid Key Cryptographic Engine (HKCE) and Memory-Based Lightweight Tokenization (MBLT).

5.4. Computational Complexity Analysis

In this subsection, we analyse the time complexity of the proposed HKCE-MBLT scheme. For a multigraph representing the size of the input data, the time complexity of the HKCE-MBLT is described by function f, where $f(n)$ represents the maximum number of steps required to solve the lightweight computational problem instance with input data of length n. Each client performs scalar multiplication operations involving $2*$ prior, $2*$ posteriors, $2*$ ECC operations, $2*$ GME operations, $2*$ Paillier encryption operations, and hash operations to encrypt the ECTS and provide a signature for payment access identity verification, especially when m clients are involved in the reputation transaction. Consequently, computational complexity of each client is $O(1)$.

Each client then combines the ciphertexts of m clients and performs $(m + 2)$ scalar multiplication operations on ECC, 1 m point addition operation on ECC, and 5* hash operations to batch-authenticate the signatures of the m clustered clients. Therefore, the computational complexity of the HKCE-MBLT increased to $O(m)$.

To determine the total number of active customers in the relevant segment of the e-commerce client list, the cloud payment gateway provider performs two scalar multiplication operations on the ECC, two Paillier decryption operations, and five hash operations. Thus, computational complexity the cloud payment gateway provider is $O(1)$. Therefore the space complexity for each client transaction, banking vendor, and payment gateway is $O(1)$, $O(1)$, $O(m)$, respectively.

Under extreme circumstances, it is impossible to determine the exact $f(n)$ complexity of an algorithm. Thus, we are satisfied with an approximation of the scalable rate of $f(n)$. The following notation is used, such that $Let\ f\ and\ g$ represent two mappings from $N\ to\ \mathcal{R}^+$.

- If there is a constant $c > 0$ such that $f(n) \leq cg(n)$ for any sufficiently large n, this is given as $f(n) = O(g(n))$.
- If there is a constant $c > 0$ such that $f(n) \geq cg(n)$ for all sufficiently big n, then $f(n) = (g(n))$
- If both $f(n)$ and $f(n)$ are equal to $O(g(n))$, then $f(n) = \Omega(g(n))$
- If $f(n) = g(n)$, then f has a scaled rate of $\Omega(g(n))$.
- If $f(n) = \theta(\mathrm{g(n)})$ or $f(n) = (g(n))$, then f has the highest or lowest complexity growth $g(n))$.

The MBLT-time KEM algorithm is said to have a complexity of $O(g(n))$ because it's temporal or spatial complexity of $O(g(n))$.

6. System Evaluation

6.1. HKCE-MBLT Security Scheme Analysis and Comparison

In this section, a thorough analysis of the proposed Hybrid Key Cryptography Engine (HKCE) integrated with the MBLT security scheme is conducted and compared with existing Public Key Cryptography (PKC) techniques under a moderate platform computing workload. The HKCE is integrated with the Performance Testing System (PTS) during the simulation setup in Section 6.3 to evaluate its efficiency. The primary goal is to understand the impact of Quality of Service (QoS) metrics on computational resources dedicated to security.

6.2. Numerical Simulation

6.2.1. Performance Evaluation of SaaS E-Transactional End-to-End Service Provisioning

This section evaluates the numerical performance of SaaS e-transactional services under the proposed PKI. The aim is to assess the efficiency of lightweight encryption and decryption, harnessing the power of HKCE-MBLT within the OSCARS platform (Figures 1 and 2). The evaluation explores the computational overhead of ECTS and its impact on Quality of Service (QoS), leveraging bilinear maps (BM) for reducing batch verification costs as detailed in reference [52]. Bilinear pairing (BP) is employed to scrutinize computational processing times across various schemes.

6.2.2. Testbed Configuration and Implementation

The ECTS-HKCE-MBLT operations were implemented within the Miracl Java SDK, integrating the M-Pin system for enhanced security on the OSCARS platform. The test environment consisted of:

- Processor: Intel Core i7-8300 GPU at 3.40 GHz
- Memory: 16 GB RAM
- Operating System: Windows 11

Each PKC computational operation was executed 100 times, and results were averaged for consistency. In terms of the Cryptographic Workflow, Key Encapsulation Mechanism (KEM) was used to transform uniform random numbers into plain text. An unbiased estimator determined decryption key lengths, and ciphertexts were generated via MD5 double encryption. Plaintext was successfully recovered once Bayesian criteria were satisfied.

6.2.3. Performance Metrics and Analysis

Recall that Tables 3 and 4 provide a probability distribution likelihood function analysis, showcasing KEM transformations and key length estimations. This transformation process validates the efficiency of HKCE-MBLT in lightweight cryptographic operations completed from Table 5. The experimental results demonstrate significant reductions in computational overhead and enhanced throughput, confirming the suitability of the proposed scheme for real-time SaaS e-transaction services. This evaluation reinforces the security and performance advantages of the ECTS-HKCE-MBLT system in addressing the challenges of modern SaaS e-commerce platforms.

Table 5. Simulation runtime results.

SN	Simulation Instance	Values
1	Total events	1526067
2	Average Speed	141696 events/sec
3	Time Elapsed	11seconds
4	Simulated	1 hr. 0 min. 0 sec
5	Simulation Log	3781 entries

6.3. Result Analysis

6.3.1. Analysis of Key Computational Parameters for HKCE-MBLT

In this section, we analyse the key computational parameters of the proposed PKI, focusing on encryption and decryption processes. The analysis incorporates Figure 2, which served as a validation context during a prior deployment phase, as documented in [53]. We then constructed a bilinear pairing for the HKY-LA and ECC key scattering AES-CBC algorithm with a key scattering scheme as a more efficient PKI. This is done with a similar security level of 80 bits [52]. Subsequently, a bilinear pairing was created for the HKY-LA and ECC key scattering AES CBC algorithms. A comparable 80-bit security level is used. The BP construct, $\hat{e}: G_1 * \dot{G}_1 \rightarrow G_T$ was adjusted. G_1 is represented as P and q by the key generator and the order respectively. G_1 has an 80-bit security level and is composed of points on the elliptic curve $\hat{E}: y^2 = x^3 + x \; Mod \; \hat{P}$. The prime numbers P and q have lengths of 512 bits and 160 bits, respectively. This is how ECC is constituted: In the equation $y^2 = x^3 + ax + b \; Mod \; p$, where P and q are two 160-bit prime numbers and $a, b \; \in Z_p^*$. G_a represents an additive group with a generator of P and a prime order q. Each of the selected schemes aims to safeguard users' and data privacy during data transmission in an open communication environment. For convenience, Table 6 is used with key symbols to display the execution time of cryptographic procedures. The selected cutting-edge techniques demonstrated the computational and communication costs of the proposed scheme. An elaborate analysis with selected metrics was used to test the performance of the system considering ECC. Ten user location sites were created with each site having random users based on their probability distributions. For these scenarios, the system attributes were configured using ECTS HTTP service and then imported after several trial runs. The validation experiment is used to show the effectiveness of the proposed scheme over four characteristically similar cryptographic schemes in common use within the industry. These are the Diffie-Hellman key exchange (DHKE) [55], Elliptic Curve Cryptography (ECC) [21], elliptic-curve Diffie–Hellman (ECDH) [18], and Digital Signature Scheme (DSS) [56].

Table 6. Operation and Execution Time for Selected Asymmetric Cryptographic Schemes

Model Notation	Explanation of Asymmetric Cryptographic Processes	HKCE-MBLT	DHKE	ECC	ECDH	DSS
T_{bp}	Duration of the bilinear pairing's execution (BP)	0.6566	3.2830	2.6264	3.750	3.8140
T_{bp}^{m}	Time it takes to perform a multiplication operation using BP	0.1600	0.4202	1.0230	2.0108	0.3782
T_{bp}^{a}	Computation time for a bilinear pairing-based point addition operation	0.0083	0.1600	0.0723	0.0928	0.01902
T_{ecc}^{m}	Duration of an elliptic curve-based scalar multiplication operation	0.0083	0.1650	0.0879	0.8021	0.0620
T_{ecc}^{a}	Time required to perform an elliptic curve-based point addition operation	0.0002	0.0025	0.0150	0.0018	0.0067
T_{h}	One-way hash operation's start time	0.0001	0.0001	0.0001	0.0001	0.0001
T_{mtp}	Hash-to-point operation's execution period	0.5400	3.2802	3.3870	3.2610	3.2001
T_{exp}	Single exponential operation's duration	0.0565	0.3340	`0.3458	0.3440	3.3847

The study rigorously evaluated key practical security metrics critical to security QoS profiling [57]. The evaluation metrics derived from the simulation engine are plotted in Figures 4, 5, 6, 7 and 8. These metrics forms the backbone for the robust security assessments summarised as follows:

- Computational Overhead: Quantifies the resource consumption or demands associated with encryption and decryption processes, ensuring efficient resource allocation without compromising security.
- Throughput: Measures the cryptographic volume data processed per unit time. This shows the ability of the system to manage high transactional scenarios securely.
- Execution Time: Assesses the operational efficiency of cryptographic algorithms for real-time transactions
- Query Response Time: Depicts the responsiveness and agility of the ECT-SaaS platform needed for guaranteeing smooth user transaction experience.
- Vulnerability Index/Weight: Evaluates the system's resilience to potential attack vectors, breaches, threats but also provides stability against cyberattacks in ECTS.

Figure 4 illustrates the computational overhead of the ECTS system using the proposed HKCE-MBLT scheme, which outperforms traditional key scattering methods in encryption efficiency. This advantage stems from the lightweight MBLT method, which eliminates reliance on legacy attributes. The proposed scheme also requires fewer computational resources for both encryption and decryption, as it leverages Bayesian natives inherent in HKCE-MBLT instead of user-specific attributes. In contrast, asymmetric encryption schemes prove unsuitable for continuous real-world security because of their significantly higher overhead.

In the proposed system, public keys encrypt data during the initial handshake, while private keys handle decryption. This method enables both parties to securely establish and exchange a new "shared key." Each client verifies the authenticity of the ECTS transaction's signature before encrypting the transaction message and sending it to the payment processor engine. This process is completing in 0.95 seconds. MBLT consolidates responses from multiple encrypted communications into a single message, which is then forwarded to the bank for the token key.

In addition, Figure 4 highlights that the proposed HKCE-MBLT scheme achieves significantly lower computation times for the encryption and decryption compared to existing schemes across various file sizes and key lengths. Specifically, HKCE-MBLT incurs only a 25% overhead when encrypting or decrypting a 4-kilobyte key length,

while the ECC key scattering scheme imposes a 75% overhead for the same key length. This advantage is particularly evident in practical big data applications hosted in the cloud. The results underscore the superior efficiency of HKCE-MBLT, especially in minimising encryption and decryption overhead. Additionally, the study evaluates computation times across a range of file sizes and key lengths, offering a comprehensive assessment of its performance.

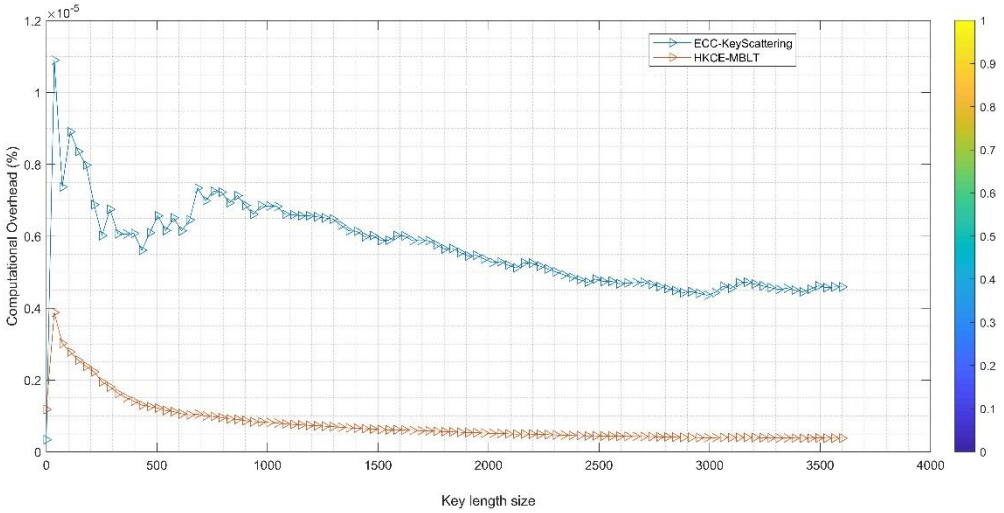

Figure 4. Computational overhead for ECTS HKCE-MBLT.

The proposed scheme significantly enhances throughput by generating a uniform ciphertext secret key of larger size. This improves encryption and decryption performance within PTS, surpassing other key-scattering schemes. Figure 5 shows that the proposed scheme achieves an asymmetric throughput of 78.57%, compared to 21.43% for key-scattering techniques. This efficiency is crucial for individual authentication and data encryption, particularly in transactional privacy controls. By integrating symmetric cryptography with Bayesian methods, the HKCE-MBLT offers improved system throughput, low hardware complexity, and effective end-to-end data encryption, reducing computational overhead and defending against various attack vectors.

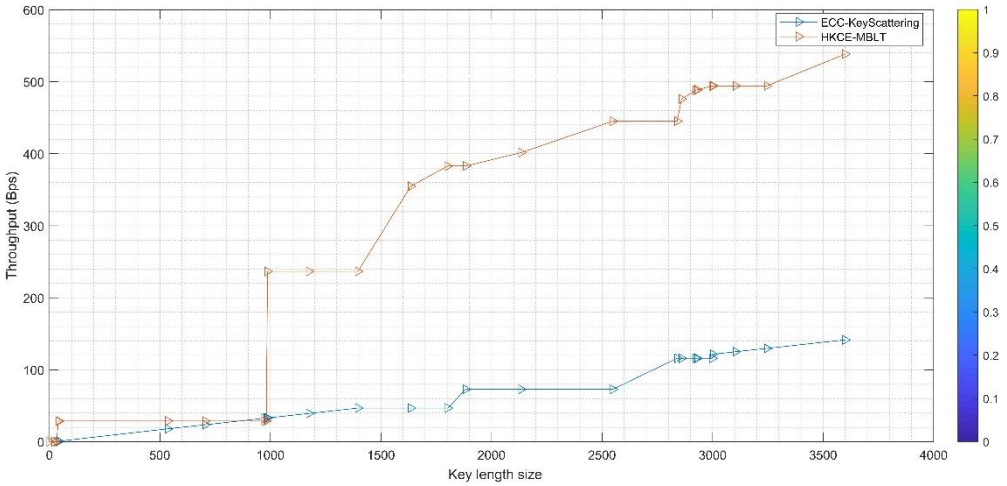

Figure 5. Transactional Throughput for HKCE-MBLT.

Figure 6 illustrates the e-transaction execution timeframe under MBLT/ HKY-LA. Our observations reveal that the proposed scheme outperforms the ECC key-scattering scheme in terms of execution efficiency during both encryption and decryption processes. The superior performance is attributed to the proposed scheme's reliance on the lightweight MBLT authorization policy, which results in lower latency (12.5%) and faster execution times. This reduction in latency is due to MBLT's minimal computational overhead during decryption. In contrast, the ECC key-scattering scheme exhibits a latency profile of 87.5%.

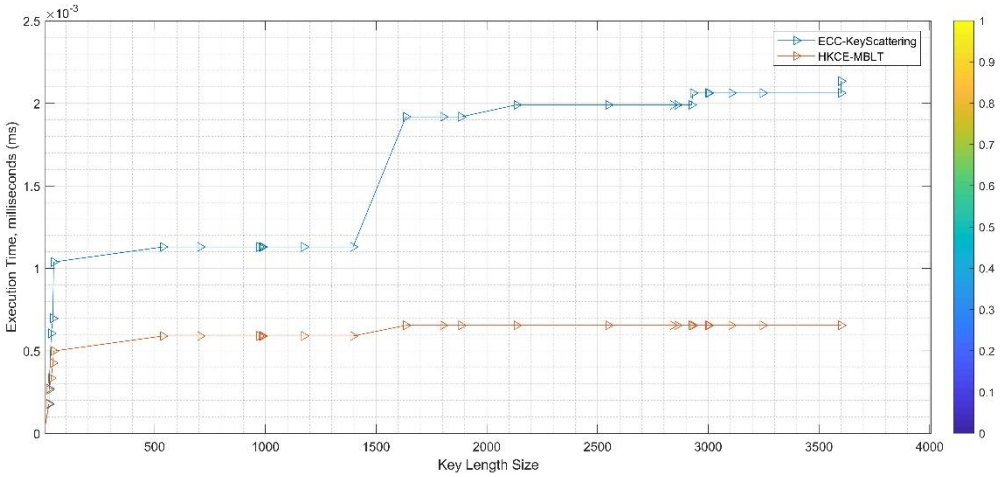

Figure 6. ECTS HKCE-MBLT Execution timeframe.

Table 7. Performance Comparison of HKCE-MBLT and ECC-Key Scattering Scheme for SaaS IoT Transaction Security. Figure 7 illustrates that the proposed HKCE-MBLT scheme offers significantly faster computation times for both encryption and decryption compared to other schemes. Additionally, the BVWI (Bandwidth Vulnerability Weight Index) measures the system's resilience against automated DoS attacks and other human-induced threats. When a vulnerability bandwidth depletion DDoS attack (VBDDA) is introduced in the ECTS, the HKCE-MBLT scheme demonstrates superior resilience with a vulnerability index of approximately 2%, compared to around 98% for ECC key-scattering modes. The constant size of the ciphertext and secret key in the proposed scheme enhances encryption and decryption efficiency, reducing bandwidth depletion, particularly during automated attacks.

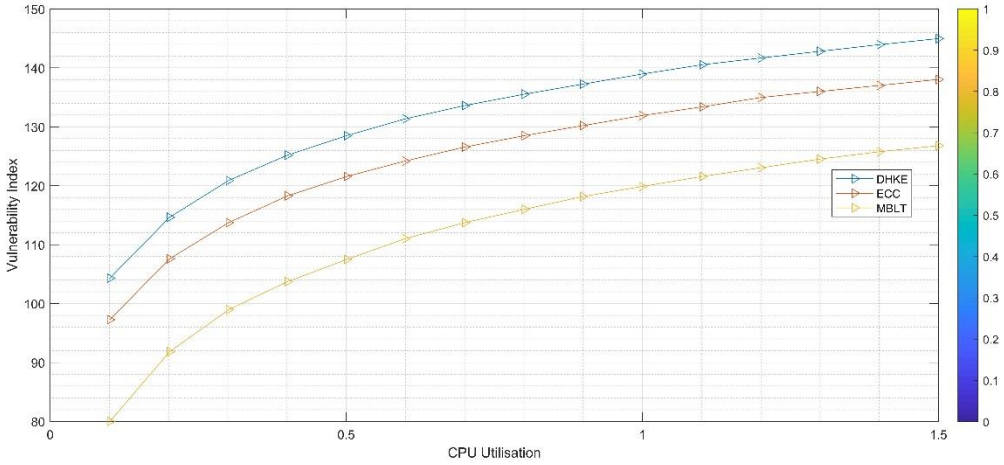

Figure 7. ECTS Vulnerability Weight Index.

From Table 7, the proposed HKCE-MBLT scheme significantly outperforms the ECC-Key Scattering Scheme (KSS) across key metrics essential for SaaS transaction security in IoT systems. With 25% computational overhead, 78.57% throughput, and 12.5% execution time, HKCE-MBLT ensures faster, more efficient processing, making it ideal for high-performance, real-time environments. Its superior 2% bandwidth vulnerability weight index and 73.53% faster authentication query responses further enhance security and system responsiveness during high transactional volumes or demands. By integrating hybrid cryptography and memory-based lightweight tokenization, HKCE-MBLT delivers scalable, secure, and resource-efficient solutions, protecting against common vulnerabilities and optimizing SaaS IoT transactions.

Table 7. Performance Comparison of HKCE-MBLT and ECC-Key Scattering Scheme for SaaS IoT Transaction Security

Attack Vector Metrics	**Proposed HKCE-MBLT (%)**	**ECC- Key Scattering Scheme (KSS) (%)**
Computational Overhead (Bytes/size)	25.00	75.00

Throughput (Bps)	78.57	21.43
Execution timeframe (secs)	12.50	87.50
Bandwidth Vulnerability Weight Index (BVWI)	2.00	98.00
Authentication Query Response (secs)	73.53	26.47

Figure 8 depicts the ECTS authentication query response under an asymmetric attack, demonstrating the efficiency of the HKCE-MBLT scheme in expediting information retrieval from the SaaS e-commerce platform. By incorporating secured load balancers on auto-scaling servers hosting SaaS applications, the proposed scheme ensures robust security while maintaining a satisfactory response time. Despite the high computational demands of e-transactional systems, the scheme excels in privacy protection, batch verification, identity privacy, efficient authentication, and transaction credibility. These results are further detailed in Table 7.

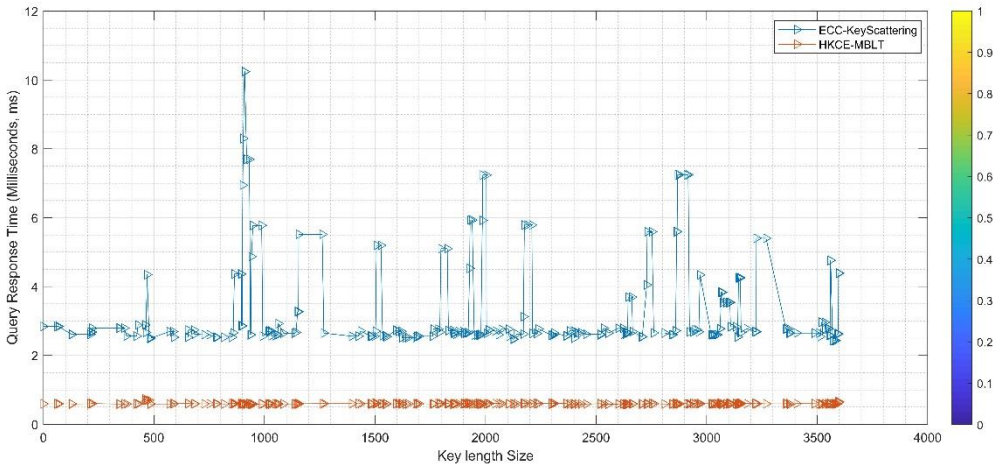

Figure 8. ECTS authentication query response.

Table 8 provides a comparative analysis of key schemes, including Diffie-Hellman Key Exchange (DHKE), Elliptic Curve Cryptography (ECC), Rivest–Shamir–Adleman (RSA), Elliptic Curve Diffie-Hellman (ECDH), Digital Signature Scheme (DSS), and Probabilistic Verifiable Visual Fingerprint (PVVF), highlighting their security and functional aspects. In contrast, Table 8 evaluates the PPVF scheme against existing systems, underscoring significant limitations. These schemes lack lightweight encryption during PC message transmission, unlike the HKCE-MBLT scheme, and depend on hybrid signature verification techniques without lightweight optimisation. Their reliance on bilinear pairings results in high computational overhead, posing challenges for efficient authentication. Among all examined methods, the HKCE-MBLT scheme stands out by reducing malicious communications through reputation-based verification of the PC transmitting the message. Additionally, Table 8 highlights the MBLT-KEM scheme's superior security and functionality compared to other existing schemes, further validating its effectiveness in addressing authentication challenges in SaaS IoT systems.

Table 8: Comparing the proposed HKCE-MBLT scheme with existing schemes (DHKE, ECC, RSA, ECDH, DSS, PVVF) based on various functionality attributes:

Functionality Attributes	Proposed HKCE-MBLT	DHKE	ECC	RSA	ECDH	DSS	PVVF
Privacy Protection	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Encryption Verification	Optimized Lightweight	Yes	Yes	Yes	Yes	Yes	Yes
Efficient Authentication	Yes	No	Yes	No	No	No	Yes
Message Credibility	High Trust (Reputation-Based)	Low	Low	Low	Low	Low	Low
Latency Profile	Low Latency	Low	High	Low	Low	Low	Low
Throughput Index	High Throughput	Low	Low	Low	Low	Low	Low

Table 8 illustrated that our proposed HKCE-MBLT scheme outperforms existing cryptographic methods such **as** DHKE**,** ECC**,** RSA**,** ECDH**,** DSS**,** and PVVF by offering efficient authentication, low latency, and high throughput with lightweight encryption verification. Unlike other schemes, HKCE-MBLT incorporates a reputation-based system for message credibility, ensuring high trust and privacy protection with minimal computational overhead. While other methods rely on computationally intensive processes and hybrid techniques, resulting in higher latency

and reduced efficiency, HKCE-MBLT is optimized for modern IoT and SaaS environments, making it the superior choice for secure, real-time communication.

6.3.2. Cryptographic Scheme Assessment

Our evaluation compares the computational efficiency of the HKCE-MBLT scheme against other cryptographic methods, particularly in encryption and decryption tasks. By utilising Figure 2, we illustrate the processing efficiencies achieved during validation and deployment, highlighting the operational advantages of HKCE-MBLT over conventional schemes. The key findings are summarised below.

1. Encryption/Decryption Performance:
    - HKCE-MBLT demonstrates reduced computational overhead compared to traditional approaches.
    - The lightweight nature of HKCE-MBLT minimises resource consumption, making it suitable for real-time e-transaction services.
2. Validation and Deployment Senarios:
    - During the previous deployment phase, Figure 2 provided critical insights into the scheme's robustness and practical applicability.
    - Our validation confirmed HKCE-MBLT's ability to sustain high throughput and low latency in dynamic SaaS environments.

By leveraging the computational strengths of HKCE-MBLT, the proposed PKI framework ensures enhanced performance and security in e-transaction systems, underscoring its effectiveness in modern cryptographic applications.

7. Conclusion and Future Work

This study demonstrated the effectiveness of HKC-MBLT in enhancing SaaS transaction security through hybrid cryptography and memory-based lightweight tokenization. The approach is based on a lightweight cryptographic key exchange mechanism (LKEM) integrated into the ECTS framework under HKCE-MBLT. We applied Hybrid Public Key Encryption (PKE) scheme for the ECTS authentication using a joint density function distribution. The PKC authentication mechanism achieves a 0.95-second server response time globally through computational analysis with a low latency threshold. The contribution of this study extends to providing an unbiased uniform estimator for service accessibility in SaaS e-transaction processes, reducing access latency for improved user experiences. Results show that the integration of a Lightweight Cryptographic Key Exchange Mechanism (KEM) into the ECTS framework significantly improves authentication processes. Empirical experiments highlight the KEM's superior performance for ECTS, showcasing attributes such as computational efficiency, throughput, execution and query response times, and the vulnerability index. The results show that the proposed scheme outperforms traditional methods such as Diffie-Hellman and RSA, particularly in terms of computational efficiency and query response times. This work sets the foundation for future research, which will explore lightweight neuromorphic processing to further strengthen encryption mechanisms, particularly for SaaS environments requiring robust, and scalable security solutions.

Disclosure statement: No potential conflict of interest was reported by the author(s).
Funding Acknowledgement: This work was supported in part by the Tertiary Education Trust Fund (Tetfund) Nigeria under the grant Number ("TETF/ES/ UNIV/IMO STATE/TSAS/2021″). Reference Project: Autonomous Driverless Cars using Resilient AI. This paper and the research behind it would not have been possible without the exceptional support of the University of Johannesburg, South Africa.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, [K.C.O], upon reasonable request.

Code Availability:
The code used to calculate cryptographic operation execution times is available at:
https://github.com/hongcheng234/cryptographic-operations.

Credit Authorship Contribution Statement
Kennedy Chinedu Okafor: Writing – review & editing, Conceptualization, Formal analysis, Validation, Supervision, Methodology, and Funding acquisition.
Omowunmi Mary Longe: Writing – review & editing, supervision.
Michael Obinna Ezeja: Writing – review & editing.
Ikechukwu Ignatius Ayogu: Writing – review & editing.
Kelvin Anoh: Writing – review & editing.
Bamidele Adebisi: Supervision, Funding acquisition, Supervision.
All authors have read and approved the final version of the manuscript

References

[1] M. Haranas, "Gartner: Top 5 Cloud SaaS, IaaS And Services In $600B Market," *CRN Online*, Apr. 28, 2023. [Online]. Available: https://www.crn.com/news/cloud/gartner-top-5-cloud-saas-iaas-and-services-in-600b-market. [Accessed: Oct. 05, 2024].

[2] P. Cohan, "Generative AI Cloud Platforms," in *Brain Rush*, Berkeley, CA: Apress, 2024, pp. 167–235. doi: 10.1007/979-8-8688-0318-5_6.

[3] S. Aleem, R. Batool, S. Alkobaisi, F. Ahmed and A. Masood Khattak, "SaaS Application Maturity Assessment Model," in *IEEE Access,* vol. 12, pp. 128305-128325, 2024, doi: 10.1109/ACCESS.2024.3455937.

[4] R. Mantu, M. Chiroiu, and C. Raiciu, "Process Identity-Based Firewalling," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds., vol. 14983, *Lecture Notes in Computer Science, Cham: Springer*, 2024, pp. 202–221. doi: 10.1007/978-3-031-70890-9_11.

[5] V. Casola, A. De Benedictis, C. Mazzocca and R. Montanari, "Designing Secure and Resilient Cyber-Physical Systems: A Model-Based Moving Target Defense Approach," *in IEEE Transactions on Emerging Topics in Computing*, 12(2), pp. 631-642, April-June 2024, doi: 10.1109/TETC.2022.3197464.

[6] Z. Rehman, I. Gondal, M. Ge, H. Dong, M. Gregory, and Z. Tari, "Proactive defense mechanism: Enhancing IoT security through diversity-based moving target defense and cyber deception," Computers & Security, vol. 139, 2024, Art. no. 103685. doi: 10.1016/j.cose.2023.103685.

[7] A. E. Adeniyi, R. G. Jimoh, and J. B. Awotunde, "A systematic review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," *Computers and Electrical Engineering*, vol. 118, Part A, 2024, Art. no. 109330. doi: 10.1016/j.compeleceng.2024.109330.

[8] M. K. H. Al-Dulaimi, A. M. Al-Dulaimi, O. M. Al-Dulaimi, A. F. Abdulqader and A. Zakharzhevskyi, "Threats in Cloud Computing System and Security Enhancement," *IEEE 2024 35th Conf. of Open Innovations Association (FRUCT), Tampere, Finland,* 2024, pp. 82-93, doi: 10.23919/FRUCT61870.2024.10516377.

[9] K. P. Kumar, B. R. Prathap, M. M. Thiruthuvanathan, H. Murthy, and V. J. Pillai, "Secure approach to sharing digitized medical data in a cloud environment," *Data Science and Management*, vol. 7, no. 2, pp. 108–118, 2024. doi: 10.1016/j.dsm.2023.12.001.

[10] Saini, H., Singh, G., Dalal, S. et al. Enhancing cloud network security with a trust-based service mechanism using k-anonymity and statistical machine learning approach. Peer-to-Peer *Netw. Appl. 2024.* https://doi.org/10.1007/s12083-024-01759-y

[11] C. Paar, J. Pelzl, and T. Güneysu, "Introduction to Public-Key Cryptography," in *Understanding Cryptography*, Berlin, Heidelberg: Springer, 2024, pp. 177–203. doi: 10.1007/978-3-662-69007-9_6.

[12] S. Mishra, S. Mishra, Y. C. Toh, S. Mishra, and P. T. Vi, "Mitigating the Threat of Multi-Factor Authentication (MFA) Bypass Through Man-in-the-Middle Attacks Using EvilGinx2," in *Cybersecurity and Privacy: Innovative Trends and Future Directions*, A. Bijalwan, R. Bennett, G. B. Jyotsna, and S. N. Mohanty, Eds. Wiley, 2024, ch. 5. doi: 10.1002/9781394272303.ch5.

[13] B. Sowmiya, E. Poovammal, K. Ramana, S. Singh and B. Yoon, "Linear Elliptical Curve Digital Signature (LECDS) With Blockchain Approach for Enhanced Security on Cloud Server," *in IEEE Access*, vol. 9, pp. 138245-138253, 2021.

[14] Amirkhanova, Dana Sairangazhykyzy, Maksim Iavich, and Orken Mamyrbayev. 2024. "Lattice-Based Post-Quantum Public Key Encryption Scheme Using ElGamal's Principles" *Cryptography* 8, no. 3: 31. https://doi.org/10.3390/cryptography8030031.

[15] G. Y. Ismail, S. Alhayali, S. W. Kareem, and Z. S. Hussain, "Secure Data in the Cloud with a Robust Hybrid Cryptographic Approach," *Journal of Electrical Systems*, vol. 20, no. 2, pp. 2450-2457, 2024.

[16] X. Zhu, Z. Di, Q. Yao, X. Dong, J. Wang and Y. Shen, "Performance-Power Tradeoff in Heterogeneous SaaS Clouds With Trustworthiness Guarantee," *in IEEE Transactions on Computers*, vol. 72, no. 6, pp. 1554-1567, 1 June 2023, doi: 10.1109/TC.2022.3214626.

[17] A. Maarouf, R. Sakr and S. Elmougy, "An Offline Direct Authentication Scheme for the Internet of Medical Things Based on Elliptic Curve Cryptography*," in IEEE Access*, vol. 12, pp. 134902-134925, 2024, doi: 10.1109/ACCESS.2024.3458424.

[18] K. H. Moussa, A. H. El-Sakka, S. Shaaban and H. N. Kheirallah, "Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange for LTE Military Grade Communication," *in IEEE Access*, vol. 10, pp. 80352-80364, 2022.

[19] M. Hegde, R. R. Rao and B. M. Nikhil, "DDMIA: Distributed Dynamic Mutual Identity Authentication for Referrals in Blockchain-Based Health Care Networks," *in IEEE Access,* vol. 10, pp. 78557-78575, 2022.

[20] K. -A. Shim, "A Survey on Post-Quantum Public-Key Signature Schemes for Secure Vehicular Communications," *in IEEE Trans. on Intelligent Transportation Systems*, 23(9), pp. 14025-14042, 2022,

[21] S. T. Bukhari, M. U. Janjua and J. Qadir, "Secure Storage of Crypto Wallet Seed Phrase Using ECC and Splitting Technique," *in IEEE Open Journal of the Computer Society*, vol. 5, pp. 278-289, 2024, doi: 10.1109/OJCS.2024.3398794.
[22] I. Kim, W. Susilo, J. Baek and J. Kim, "Harnessing Policy Authenticity for Hidden Ciphertext Policy Attribute-Based Encryption," *in IEEE Transactions on Dependable and Secure Computing*, 19(3), pp. 1856-1870, 1 2022.
[23] Z. Wang et al., "Efficient Location-Based Skyline Queries With Secure R-Tree Over Encrypted Data," in *IEEE Transactions on Knowledge and Data Engineering,* vol. 35, no. 10, pp. 10436-10450, 1 Oct. 2023, doi: 10.1109/TKDE.2023.3253883.
[24] X. Li et al., "Cyber-Physical Power Systems: Exploring a Streamlined Signcryption Scheme for Resource-Limited Smart Terminals," *in IEEE Transactions on Industrial Informatics*, vol. 20, no. 7, pp. 9749-9760, July 2024, doi: 10.1109/TII.2024.3379640.
[25] A. Aljuhani, A. Alamri, P. Kumar and A. Jolfaei, "An Intelligent and Explainable SaaS-Based Intrusion Detection System for Resource-Constrained IoMT," *in IEEE Internet of Things Journal,* vol. 11, no. 15, pp. 25454-25463, 1 Aug.1, 2024, doi: 10.1109/JIOT.2023.3327024.
[26] S. Zhang, S. Ray, R. Lu, Y. Zheng, Y. Guan and J. Shao, "Towards Efficient and Privacy-Preserving Interval Skyline Queries Over Time Series Data," *in IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1348-1363, 1 March-April 2023, doi: 10.1109/TDSC.2022.3153759.
[27] R. R. Irshad *et al*., "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," *in IEEE Access,* vol. 11, pp. 105479-105498, 2023, doi: 10.1109/ACCESS.2023.3318755.
[28] S. Pfeiffer and N. Tihanyi, "D(HE)at: A Practical Denial-of-Service Attack on the Finite Field Diffie–Hellman Key Exchange," in *IEEE Access,* vol. 12, pp. 957-980, 2024, doi: 10.1109/ACCESS.2023.3347422.
[29] H. Wang, J. Wen, J. Liu and H. Zhang, "ACKE: Asymmetric Computing Key Exchange Protocol for IoT Environments," *in IEEE Internet of Things Journal,* vol. 10, no. 20, pp. 18273-18281, 15 Oct.15, 2023, doi: 10.1109/JIOT.2023.3279283.
[30] X. Zhang, K. Chen, J. Ding, Y. Yang, W. Zhang and N. Yu, "Provably Secure Public-Key Steganography Based on Elliptic Curve Cryptography," in *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3148-3163, 2024, doi: 10.1109/TIFS.2024.3361219.
[31] D. S. Gupta, "PiLike: Post-Quantum Identity-Based Lightweight Authenticated Key Exchange Protocol for IIoT Environments*," in IEEE Systems Journal,* vol. 18, no. 1, pp. 15-23, March 2024, doi: 10.1109/JSYST.2023.3335217.
[32] Y. Guo and Y. Guo, "CS-LAKA: A Lightweight Authenticated Key Agreement Protocol With Critical Security Properties for IoT Environments," *in IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 4102-4114, Nov.-Dec. 2023, doi: 10.1109/TSC.2023.3309860.
[33] R. Subrahmanyam, N. R. Rekha and Y. V. S. Rao, "Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme," *in IEEE Access,* vol. 11, pp. 45243-45254, 2023, doi: 10.1109/ACCESS.2023.3274468.
[34] S. Shaw and R. Dutta, "Forward Secure Offline Assisted Group Key Exchange From Isogeny-Based Blinded Key Encapsulation Mechanism," *in IEEE Transactions on Information Theory,* vol. 69, no. 7, pp. 4708-4722, July 2023, doi: 10.1109/TIT.2023.3260005.
[35] H. Y. Adarbah, M. F. Moghadam, R. L. R. Maata, A. Mohajerzadeh and A. H. Al-Badi, "Security Challenges of Selective Forwarding Attack and Design a Secure ECDH-Based Authentication Protocol to Improve RPL Security," *in IEEE Access,* vol. 11, pp. 11268-11280, 2023, doi: 10.1109/ACCESS.2022.3221434.
[36] X. Gong et al., "Defense-Resistant Backdoor Attacks Against Deep Neural Networks in Outsourced Cloud Environment*," in IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2617-2631, Aug. 2021, doi: 10.1109/JSAC.2021.3087237.
[37] M. B. Muzammil, M. Bilal, S. Ajmal, S. C. Shongwe and Y. Y. Ghadi, "Unveiling Vulnerabilities of Web Attacks Considering Man in the Middle Attack and Session Hijacking," *in IEEE Access,* vol. 12, pp. 6365-6375, 2024, doi: 10.1109/ACCESS.2024.3350444.
[38] H. Guo, J. Sun and Z. -H. Pang, "Analysis of Replay Attacks With Countermeasure for State Estimation of Cyber-Physical Systems," *in IEEE Transactions on Circuits and Systems* II: Express Briefs, vol. 71, no. 1, pp. 206-210, Jan. 2024, doi: 10.1109/TCSII.2023.3302151.
[39] Yingjiu Li and Xinwen Zhang, "A security-enhanced one-time payment scheme for credit card," *14th IEEE Int'l Workshop Research Issues on Data Eng: Web Services for e-Commerce and e-Government Applications*, *Proceedings*., Boston, MA, USA, 2004, pp. 40-47, doi: 10.1109/RIDE.2004.1281701.
[40] L. Dickson and R. Soderstrom, "Lasers in supermarket point of sale systems," *in IEEE Journal of Quantum Electronics*, vol. 11, no. 9, pp. 845-846, September 1975, doi: 10.1109/JQE.1975.1068888.

[41] J. Huang et al., "KeystrokeSniffer: An Off-the-Shelf Smartphone Can Eavesdrop on Your Privacy From Anywhere," in *IEEE Transactions on Information Forensics and Security, vol*. 19, pp. 6840-6855, 2024, doi: 10.1109/TIFS.2024.3424301.
[42] F. Castaño, E. F. Fernañdez, R. Alaiz-Rodríguez and E. Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," *in IEEE Access,* vol. 11, pp. 40779-40789, 2023, doi: 10.1109/ACCESS.2023.3268027.
[43] A. Kumar, G. Somani and M. Agarwal, "Comparing HAProxy Scheduling Algorithms During the DDoS Attacks," *in IEEE Networking Letters*, 6(2), pp. 139-142, 2024, doi: 10.1109/LNET.2024.3383601.
[44] J. Li, D. Yu, W. Ma, J. J. R. Liu and Y. -J. Liu, "Cooperative Control of Air–Ground Swarms Under DoS Attacks via Cloud–Fog Computing," *in IEEE Transactions on Network Science and Engineering*, vol. 11, no. 5, pp. 4278-4292, Sept.-Oct. 2024, doi: 10.1109/TNSE.2024.3409900.
[45] T. Yang, Y. Qiao, and B. Lee, "Towards trustworthy cybersecurity operations using Bayesian Deep Learning to improve uncertainty quantification of anomaly detection," *Computers & Security*, vol. 144, 2024, Art. no. 103909. doi: https://doi.org/10.1016/j.cose.2024.103909.
[46] "MATLAB." *MathWorks*. [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 06-Oct-2024].
[47] Minitab 19.2." *Minitab*. [Online]. Available: http://www.minitab.com/en-us/products/minitab. [Accessed: 06-Oct-2024].
[48] F. Mohamed, B. AlBelooshi, K. Salah, C. Y. Yeun and E. Damiani, "A Scattering Technique for Protecting Cryptographic Keys in the Cloud. *IEEE 2nd Int' Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pp. 301-306, 2017.
[49] M. Bottarelli, P. Karadimas, G. Epiphaniou, D. K. B. Ismail and C. Maple, "Adaptive and Optimum Secret Key Establishment for Secure Vehicular Communications," *in IEEE Transactions on Vehicular* Technology, 70(3), pp. 2310-2321, 2021.
[50] K. -c. Li, P. -b. Wang and R. -h. Shi, "A Novel Privacy-Preserving Range Query Scheme with Permissioned Blockchain for Smart Grid," *in IEEE Transactions on Emerging Topics in Computing*, doi: 10.1109/TETC.2024.3386803.
[51] S. Schäge, "New Limits of Provable Security and Applications to ElGamal Encryption," in *Advances in Cryptology – EUROCRYPT 2024*, M. Joye and G. Leander, Eds., vol. 14654, *Lecture Notes in Computer Science*. Cham: Springer, 2024. doi: https://doi.org/10.1007/978-3-031-58737-5_10.
[52] H. Cheng, M. Shojafar, M. Alazab, R. Tafazolli and Y. Liu, "PPVF: Privacy-Preserving Protocol for Vehicle Feedback in Cloud-Assisted VANET," *in IEEE Transactions on Intelligent Transportation Systems*, 23(7), pp. 9391-9403, 2022.
[53] K. C. Okafor *"*Development of a Model for Smart Green Energy Management Using Distributed Cloud Computing Network*", Ph.D. Thesis, University of Nigeria Nsukka, 2017.*
[54] L. Xiao, N. Tuya and R. Wu, "Application of Computer Database Signature Encryption Algorithm in Cross-Border E-Commerce Payment and Settlement System," *IEEE ICIICS, Kalaburagi, India, 2023*, pp. 1-6, doi: 10.1109/ICIICS59993.2023.10421299.
[55] A. C. -C. Yao and Y. Zhao, "Privacy-Preserving Authenticated Key-Exchange Over Internet," *in IEEE Transactions on Information Forensics and Security,* 9(1), pp. 125-140, Jan. 2014, doi: 10.1109/TIFS.2013.2293457.
[56] M. Bisheh-Niasar, R. Azarderakhsh and M. Mozaffari-Kermani, "Cryptographic Accelerators for Digital Signature Based on Ed25519," *in IEEE Transactions on Very Large Scale Integration (VLSI)* Systems, 29(7), pp. 1297-1305, 2021.
[57] F. N. Ugwoke, K. C. Okafor and V. C. Chijindu, "Security QoS profiling against cyber terrorism in airport network systems," *2015 International Conference on Cyberspace (CYBER-Abuja)*, Abuja, Nigeria, 2015, pp. 241-251, doi: 10.1109/CYBER-Abuja.2015.7360516.