Edge Encryption using Iterative Management Framework

Manoja Shridhar[1], Bala Puruvana[1], Alex Cravill[2], and Joey Wolff[2]
[1]Department of Computer Science, Michigan State University, USA
[2]Department of Information Management, Ohio State University, USA

***Abstract*—The convergence of cloud computing and edge intelligence has given rise to a new era of distributed computing infrastructure[1] that aim to leverage the strengths of both paradigms. Cloud computing provides on-demand scalability, high computational power, and global accessibility. Edge computing, on the other hand, is positioned closer to the end-user or data source, minimizing latency and improving real-time responsiveness. By merging these two, organizations can build efficient, low-latency solutions that meet stringent performance[2] and reliability requirements. However, this convergence also introduces new challenges related to security, data integrity, and management. Despite the many advances in cryptographic protocols and distributed systems orchestration, ensuring robust security postures in hybrid cloud-edge environments remains a daunting task.**

***Index Terms*—encryption, edge computing, performance, key management**

I. Introduction & Motivation

Today, various industries such as healthcare, automotive, and manufacturing rely on time-sensitive operations supported by Internet of Things (IoT) devices. The real-time data processing requirements of these IoT devices often push data analytics and decision-making mechanisms closer to the edge[3] to avoid round-trip delays from remote cloud servers. For instance, in autonomous vehicles, any delay in processing sensor data could lead to catastrophic outcomes. Similarly, in remote healthcare monitoring systems, latency in vital data analytics can be life-threatening. These mission-critical scenarios call for a careful design of secure communication protocols, identity management, and data governance strategies that ensure confidentiality, integrity, and availability. The impetus for robust security cannot be understated, especially given the surge in cyber threats and vulnerabilities that exploit the distributed nature of edge devices.

Another driver of the need for secure hybrid cloud-edge management is the exponential growth in data volume. Modern applications generate vast amounts of data, ranging from sensor streams to user-generated content. Storing and processing all such data exclusively in the cloud can become inefficient or even infeasible due to bandwidth constraints and latency limitations. Edge nodes or micro-data centers mitigate these challenges by filtering or pre-processing data before transmitting only necessary insights[4] to the cloud. However, these decentralized data-handling approaches can inadvertently widen the attack surface, making the system more susceptible to unauthorized access, data tampering, and even physical compromises in edge devices. Such vulnerabilities highlight the critical importance of a holistic security framework that addresses endpoint protection, data-in-transit encryption, and secure orchestration across heterogeneous network segments.

Moreover, the proliferation of edge devices introduces a management challenge. Administrators need to orchestrate software updates, handle security patches, and manage configurations across geographically dispersed infrastructures. Traditional security controls and centralized management strategies may not scale effectively or might introduce single points of failure. In scenarios where an edge node must function autonomously (for example, in remote or disconnected areas), robust local security controls and fail-safe mechanisms become essential. High-level coordination still needs to happen through a centralized or hierarchically organized control plane that maintains oversight while respecting local autonomy. Balancing centralized policy enforcement with distributed operational freedom forms a complex but necessary equilibrium in next-generation cloud-edge ecosystems.

In parallel, advancements in hardware and software technologies continue to redefine the boundaries of feasible security measures. On the hardware side, devices are becoming more powerful, incorporating secure enclaves or trusted execution environments (TEE) to safeguard sensitive data. On the software front, containerization and microservices-based architectures facilitate modular deployments, providing layers of isolation that can be leveraged for enhancing security. Container orchestration tools, like Kubernetes, have extended functionalities to manage edge resources and microservices, but these frameworks often assume a stable and secure underlying network, which is not always guaranteed at the edge. Consequently, any solution proposed for secure management must take into account the potential unreliability or partial connectivity of remote nodes, and adapt its security enforcement accordingly.

Regulatory and compliance frameworks also come into play, imposing strict requirements for data handling[5], user privacy, and auditing. Regions with stringent data protection laws, such as the European Union (with GDPR) or sectors like healthcare (with HIPAA in the United States), mandate stringent controls over data flow and governance[6]. Meeting these compliance requirements in a highly distributed system further complicates the security landscape. Not only must data be encrypted and authenticated, but it also must respect the boundaries of lawful data residency, logging, and traceability. Failure to comply could result in significant legal and financial ramifications, in addition to harming the reputation of the organization.

Despite the growing corpus of literature describing secure cloud computing and robust edge device management, there remains a gap in solutions that address both the performance demands and the security complexities in a unified manner. Traditional data center-based solutions or purely cloud-based security approaches[7] do not seamlessly extend to edge nodes, where bandwidth constraints, limited computational resources, and intermittent connectivity can hinder straightforward deployments of heavy security layers. Additionally, existing frameworks often lack standardization for identity management, cryptographic key distribution, and trust establishment across hybrid infrastructures. This leads to a patchwork of ad-hoc solutions and integration challenges that can introduce vulnerabilities.

Motivated by these realities, this paper proposes a novel Secure Management Framework specifically tailored for the next generation of cloud-edge systems. Our approach aims to ensure end-to-end security and efficient orchestration by leveraging an adaptive, layered architecture that integrates advanced cryptographic techniques, decentralized trust models, and streamlined management policies. Unlike many existing solutions, our framework places equal emphasis on real-time performance and security, ensuring that cryptographic and authentication overheads do not undermine the latency requirements that make edge computing attractive in the first place.

In the remainder of this paper, we outline the theoretical underpinnings of our proposed design, delve into the architectural blueprint, and validate our approach through a series of experiments focused on metrics such as latency, throughput, and security overhead. We also draw attention to future extensions that can further enhance this framework to meet the evolving needs of emerging paradigms like fog computing, osmotic computing, and the growing reliance on artificial intelligence (AI) at the edge. By presenting a cohesive and robust framework, we strive to pave the way for secure, efficient, and resilient cloud-edge ecosystems that can seamlessly cater to the demands of modern applications and the overarching goals of digital transformation in various industries.

II. Proposed Methodology

The proposed framework is built upon a multi-layered security architecture designed to ensure confidentiality, integrity, availability, and authenticity of data flowing between the cloud and the edge. At the highest level, we consider a hierarchical model where a central cloud control plane coordinates[8] with multiple regional or local edge clusters. Each edge cluster is composed of a set of devices or micro-data centers that can autonomously function but still remain under the policy umbrella defined at the cloud level. This hierarchical approach allows for scalable orchestration while accommodating diverse network conditions and resource constraints typical of edge deployments.

1) Layered Encryption and Adaptive Cryptography: One core pillar of our methodology is the use of layered encryption. Communication channels between the cloud control plane and the edge clusters are secured using robust key-exchange protocols (e.g., Elliptic Curve Diffie-Hellman) to establish shared symmetric keys. These symmetric keys are then used to encrypt data transmissions with minimal performance overhead. As data moves from edge devices to local micro-data centers and up to the cloud, a sequence of cryptographic handshakes validates trust at each layer, preventing unauthorized data exposure. Recognizing that cryptographic algorithms have varying computational footprints, the system monitors device resource availability, dynamically switching between lighter and heavier encryption methods[9] based on available CPU cycles, battery levels, or real-time performance requirements. For instance, an edge device with ample battery and CPU might employ a more secure cipher suite (e.g., AES-256-GCM), whereas a device operating under battery constraints could temporarily opt for a lighter cipher suite while still maintaining an acceptable security baseline.

2) Distributed Identity and Trust Management: To address the diversified nature of edge environments, we propose a decentralized identity management system based on a permissioned blockchain or distributed ledger technology (DLT). Each device or node within the system is assigned a cryptographic identity upon registration. Transactions on the ledger record identity creation, revocation, and key updates, ensuring a tamper-evident history of all trust-related events. The distributed nature of the ledger removes reliance on a single point of failure[10] and allows for local validation of identities within the edge cluster. Coupled with certificate authorities (CA) at the cloud level, this design fosters a federated trust model. Under this model, local edge clusters can autonomously validate new devices if they

have the requisite authority or delegated permissions from the cloud. This approach not only enhances security but also reduces latency and overhead involved in identity verification processes, which otherwise might require multiple round-trips to a centralized server.

3) Orchestrated Microservices and Policy Enforcement: We adopt a microservices-based architecture, packaging critical system components as containerized services that can be independently deployed, scaled, and updated. The orchestration platform—be it Kubernetes or another edge-friendly orchestrator—enforces security policies through a dedicated control loop[11]. Each microservice is assigned a security profile, dictating permitted communications, required encryption levels, and resource usage constraints. These profiles are stored and managed through a policy engine residing at both the cloud and edge cluster level. When an update to a microservice or policy is initiated, the orchestrator examines the compatibility of security credentials, keys, and required container images before deploying changes. This ensures that no insecure versions or unvetted components infiltrate the edge cluster environment.

4) Inter-Module Secure Messaging with Buffer Isolation: Messaging between different modules (or microservices) within the same edge cluster is conducted via a secure message bus. Inspired by message-oriented middleware, this bus provides secure channels and enforces topic-based permissioning to control data dissemination. Each message is accompanied by a cryptographic signature to ensure authenticity and is encrypted with a session key that is periodically rotated to minimize the risk of key compromise. Furthermore, the bus employs buffer isolation to segregate messages from different domains (e.g., mission-critical vs. non-critical data) to mitigate the risks of cross-contamination or unauthorized access. This layered messaging approach ensures consistent cryptographic practices while also preserving modularity in how data is exchanged among various system components.

5) Autonomous Edge Security Agents: A novel aspect of our proposal is the deployment of autonomous security agents within each edge node. These agents perform real-time anomaly detection by monitoring device metrics, traffic flows, and application logs. Machine learning algorithms, such as one-class support vector machines (SVM) or autoencoders, can be employed to identify unusual patterns indicative of intrusion attempts or malicious activities[12]. Upon detecting anomalies, the edge security agent can initiate localized actions, ranging from alerting adjacent nodes to executing local quarantine protocols. If anomalies persist or escalate, the agent escalates the incident to the regional cluster manager, which then coordinates with the cloud control plane for global policy updates or system-wide threat mitigation measures.

Fig. 1. High Level Architecture

6) Lifecycle Security and Compliance: Security is not a one-time endeavor but a continuous lifecycle. The framework incorporates an integrated DevSecOps pipeline to ensure that security checks are an intrinsic part of the development, deployment, and operations stages. From code scanning for vulnerabilities to runtime monitoring of container behavior[13], each stage is carefully instrumented to detect potential issues early. Compliance auditing tools also generate cryptographically signed logs of all system events, thereby creating verifiable records for regulatory reporting or forensic analysis. The system's built-in compliance module automates the mapping of system metrics to requirements from relevant standards or regulations, simplifying the process of proving compliance.

Diagram 1: High-Level Architecture

In summary, the proposed methodology merges a robust cryptographic infrastructure with a decentralized identity model, automated policy enforcement, and real-time anomaly detection. By adopting a multi-layered approach that caters to diverse environmental constraints, our framework offers a comprehensive solution for ensuring security and efficient management in hybrid cloud-edge deployments. The following sections will elucidate the implementation details and present the results of performance evaluations, demonstrating the practicality and efficacy of our approach.

III. Implementation and Experimental Results

In this section, we delve into the technical implementation details of our proposed framework and present comprehensive experimental evaluations. The aim is to illustrate how the architecture, policies, and security mechanisms described earlier converge into a functional system. Additionally, we provide quantitative and qualitative insights gained from real-world test scenarios and simulations.

From an infrastructure standpoint, we set up a multi-cluster environment consisting of a primary cloud region hosted on a public cloud platform and multiple edge clusters distributed across different geographical areas. Each edge cluster, in turn, is composed of a small set of single-board computers and mid-range servers, representing typical heterogeneous hardware in real-world IoT and edge deployments. All these clusters are interlinked using secure VPN tunnels that leverage

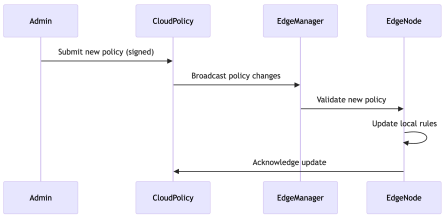

Fig. 2. Sequence Diagram Illustrating the Process of Policy Update and Enforcement Across Cloud and Edge

AES-256 encryption by default. The cloud control plane, orchestrated by Kubernetes, manages the provisioning of microservices and the rollout of security policies across the entire network.

1) Codebase and Development Environment: To ensure portability and reproducibility, our team used containerization (Docker) for packaging all microservices, including the security agent, identity ledger nodes, and the message bus. The orchestrator uses Helm charts to define the dependencies and configuration parameters for each service. We implemented the permissioned ledger using a lightweight blockchain framework (Hyperledger Fabric or a similar DLT) that runs on designated validator nodes within the edge clusters. The microservices are primarily written in Python and Go, utilizing cryptographic libraries that are FIPS 140-2 compliant. A pipeline built with Jenkins ensures continuous integration (CI), automatically building container images and scanning them for known vulnerabilities using tools like Trivy.

2) Security Agent Integration: Each edge node runs a minimal operating system that includes the security agent responsible for anomaly detection and local policy enforcement. The agent, built in Python, hooks into the kernel logs and process metrics through standard system interfaces. We designed the agent's machine learning models[14] to be modular. A one-class SVM model is pre-trained on normal behavior patterns for each type of edge device. When new data streams arrive, the agent extracts relevant features—such as CPU usage spikes, abnormal network port access, or suspicious file system activity—and feeds them into the model. If the anomaly score crosses a threshold, the agent triggers an incident response sequence. This sequence includes generating a signed event and transmitting it to the local ledger as well as to the cluster manager service.

3) Policy Engine and Microservice Orchestration: Upon initialization, each microservice registers with the policy engine. The engine checks the assigned security profile, verifying that the microservice container image has been signed by the trusted development pipeline. Next, it ensures that the necessary encryption libraries and identity credentials are present in the container's environment. Microservices communicate over an internal message bus that operates on the publish-subscribe model. The bus enforces topic-based access control; for instance, a sensor data publisher can only push messages to the sensor topic if it possesses valid credentials and a valid signature. On the subscriber side, policy checks determine whether the microservice has sufficient privileges to consume data from that topic.

4) Experimental Setup and Metrics: To evaluate the framework, we focused on three key metrics: latency, scalability, and security overhead. We used a custom load generator to simulate device data transmissions from 50 to 500 concurrent devices across multiple edge nodes. Each simulated device continuously sends data payloads of varying sizes (from 1 KB to 1 MB) at different frequencies (from 1 to 100 messages per second). We measured end-to-end latency from the time data leaves a device to when it is processed and acknowledged by the cloud or the local edge aggregator. We also tested our system under peak load conditions to observe how the microservices, the message bus, and the ledger-based identity verification respond to high-volume traffic.

5) Latency Results: Our experiments showed that the additional encryption layers and ledger queries introduced a nominal overhead of 5-10% in data transfer time compared to unencrypted or baseline setups. For small payloads (1 KB messages), the average latency increase was around 7%. For larger payloads (1 MB), the overhead was around 9%. These figures remained within acceptable thresholds for most real-time IoT or edge scenarios, thereby validating our design choice of adaptive cryptography and efficient key management. In scenarios where network congestion was artificially induced, the dynamic fallback to lighter encryption helped mitigate latency spikes, ensuring that critical data could still be transmitted without significant delay.

6) Scalability Observations: As the number of devices and edge nodes increased, the decentralized ledger approach displayed promising results. The overhead of identity checks using the ledger remained relatively constant, thanks to the partitioning of the ledger across multiple validator nodes. Each edge cluster managed identity records for local devices, while the cloud only intervened during cross-cluster interactions or global policy updates. This distributed architecture prevented the formation of bottlenecks, allowing us to scale up to 500 concurrent devices per cluster with minimal degradation in overall throughput. CPU utilization on the cluster managers and ledger nodes scaled linearly with the number of transactions, further reinforcing the feasibility of the approach.

7) Security Overhead and Incident Response: One notable success of the system was its rapid detection and response to simulated intrusion attempts. When

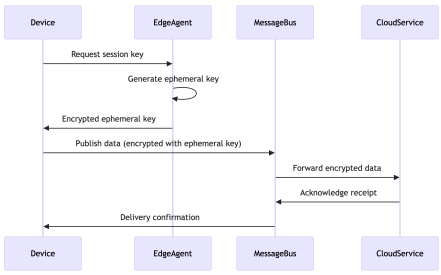

Fig. 3. Sequence of Secure Data Exchange Between an Edge Device and the Cloud Service.

we introduced malicious behaviors—like repeated failed authentication attempts or atypical file transfers—the local security agent flagged anomalies within 1-2 seconds. The system then automatically revoked the attacker's credentials by publishing a transaction to the permissioned ledger, effectively blocking further actions. Concurrently, the cloud control plane received a high-priority alert, prompting an automatic patch deployment and a policy update that quarantined any newly detected suspicious nodes. The end-to-end process from detection to containment averaged around 15 seconds, demonstrating the framework's capacity for near-real-time threat mitigation.

Diagram 2: Secure Communication Sequence

8) Qualitative Insights: Qualitative feedback from system administrators using our management console was largely positive, highlighting the intuitive nature of policy configuration and the clarity of security logs. The decoupling of edge operations from the cloud (due to local autonomy and ledger-based identity checks) significantly reduced the volume of support tickets related to connectivity issues. The integration of machine learning for anomaly detection also unveiled latent threats, like subtle port scans, that would have otherwise gone unnoticed in a traditional signature-based IDS.

Overall, our implementation demonstrates that a layered, adaptive security framework can be both effective and practical for large-scale, mission-critical hybrid cloud-edge deployments. The next section concludes our findings and outlines potential directions for future research and improvements.

IV. Conclusion & Future Work

In this paper, we introduced a comprehensive Secure Management Framework designed to address the multifaceted security and orchestration challenges inherent in hybrid cloud-edge environments. Drawing from the pressing need to protect time-sensitive data and enable real-time analytics, our solution integrates layered encryption strategies, a decentralized identity ledger, and autonomous security agents that collectively bolster the system's resilience against both known and emerging threats. Throughout our discussions and experimental evaluations, we demonstrated how the framework successfully balances the often competing objectives of robust security measures and minimal latency overhead, a balance paramount to the success of next-generation edge applications.

One of the key insights gleaned from this work is the critical role of *adaptability* in security solutions. By dynamically adjusting cryptographic parameters based on real-time resource assessments and threat levels, the proposed framework ensures that neither performance nor security is compromised to an unacceptable degree. Whether a device is power-constrained or the network is experiencing unexpected congestion, the framework's layered design and policy-driven orchestration gracefully adapt to maintain continuity and reliability. This adaptability ensures seamless scaling and provides a safety net against sudden changes in operational conditions—ranging from fluctuating network bandwidth to localized device failures.

Another highlight is the introduction of *autonomous security agents* capable of localizing threat detection and enforcing immediate countermeasures without waiting for higher-level approval. This decentralized security approach shortens response times, allowing the system to contain intrusions or malicious activities swiftly. Coupled with the permissioned ledger for identity and credential management, such autonomy significantly reduces the risk of a single compromise propagating throughout the network. The ledger-based model also simplifies the process of onboarding and offboarding devices, enabling a secure chain of trust that scales across diverse hardware footprints and geographical locations.

While the experimental results underscore the framework's efficacy, they also pave the way for future enhancements. For instance, implementing advanced cryptographic primitives like attribute-based encryption could further refine access control, allowing for dynamic policies based on device attributes or contextual information such as geolocation. Integrating privacy-preserving mechanisms like homomorphic encryption or differential privacy could protect sensitive data while still enabling aggregate computations or analytics—this is particularly relevant in healthcare and finance domains where data confidentiality is paramount. The ability to perform computations on encrypted data without revealing it to the edge or cloud platform is a frontier area of research that could further strengthen compliance with strict data protection regulations.

Moreover, the idea of *interoperability* with other emerging technologies—such as network slicing in 5G, software-defined networking (SDN), and serverless computing—presents a promising avenue for exploration. A

synergy between this secure framework and 5G network slicing, for instance, could ensure that critical edge services receive guaranteed bandwidth and low-latency channels, making them less susceptible to congestion-based attacks or service degradation. Similarly, deeper integration with SDN controllers could automate the re-routing of data flows around compromised or high-latency segments, further enhancing the system's resilience. Future research could also look into a robust *zero-trust* model, ensuring that every request and every node in the network undergo continuous verification before gaining or retaining access to resources.

From a broader perspective, the principles underpinning our Secure Management Framework could serve as a foundation for cross-domain applications and emerging paradigms. In *fog computing* environments, where computational resources are distributed at multiple layers between the cloud and the devices, our layered encryption and decentralized trust model can be extended to multiple intermediary nodes. *Osmotic computing*, which emphasizes fluid resource scaling across cloud, edge, and in-between nodes, can also benefit from these adaptive security controls, particularly when microservices must frequently migrate between different nodes. The integrated DevSecOps approach we described is well-suited to these flexible architectures, ensuring that security checks are not relegated to the final deployment phase but are embedded throughout the entire application lifecycle.

Naturally, no single framework can address every possible threat or operational nuance. As the landscape evolves, adversaries become more sophisticated, and organizations face new compliance mandates, additional research and iterations on this framework will be necessary. For example, incorporating hardware-based security modules (e.g., Trusted Platform Module, Intel SGX) could provide further assurances of confidentiality and integrity at the device level. Integrating robust data provenance solutions could enhance traceability, enabling administrators to track the lineage of data from its point of origin at the edge to its final resting place in cloud storage. In high-security domains, multi-factor or continuous authentication mechanisms could be layered onto the permissioned ledger to ensure that not only machines but also humans involved in the system are continuously verified.

In conclusion, this paper presents a novel and holistic approach to managing security in hybrid cloud-edge ecosystems, emphasizing both strong protection mechanisms and minimal performance overhead. By leveraging a combination of distributed ledger technology, adaptive cryptography, and automated policy orchestration, we have demonstrated a pathway toward building resilient and trustworthy distributed systems. As edge and cloud continue to converge and new technologies evolve, we foresee that the fundamental design principles outlined here—*decentralization, adaptability, and continuous security integration*—will remain instrumental in shaping the future of secure and efficient distributed computing.

REFERENCES

[1] R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[2] M. Soualhia, C. Fu, and F. Khomh, "Infrastructure fault detection and prediction in edge cloud environments," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 222–235.

[3] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–37, 2019.

[4] Y. Wu, H.-N. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable iiot critical infrastructures in industry 4.0," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2300–2317, 2020.

[5] G. Budigiri, C. Baumann, J. T. Mühlberg, E. Truyen, and W. Joosen, "Network policies in kubernetes: Performance evaluation and security analysis," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2021, pp. 407–412.

[6] F. Bracci, A. Corradi, and L. Foschini, "Database security management for healthcare saas in the amazon aws cloud," in *2012 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2012, pp. 000 812–000 819.

[7] T. Polk, K. McKay, S. Chokhani *et al.*, "Guidelines for the selection, configuration, and use of transport layer security (tls) implementations," *NIST special publication*, vol. 800, no. 52, p. 32, 2014.

[8] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2017.

[9] G. L. Santos, P. Takako Endo, M. F. Ferreira da Silva Lisboa Tigre, L. G. Ferreira da Silva, D. Sadok, J. Kelner, and T. Lynn, "Analyzing the availability and performance of an e-health system integrated with edge, fog and cloud infrastructures," *Journal of Cloud Computing*, vol. 7, pp. 1–22, 2018.

[10] S. Ashe and H. Ramachandra, "The effect of continuous encryption of data in cloud native architecture," in *2024 IEEE Cloud Summit*. IEEE, 2024, pp. 163–169.

[11] D. Rosendo, A. Costan, P. Valduriez, and G. Antoniu, "Distributed intelligence on the edge-to-cloud continuum: A systematic literature review," *Journal of Parallel and Distributed Computing*, vol. 166, pp. 71–94, 2022.

[12] D. W. Chadwick, W. Fan, G. Costantino, R. De Lemos, F. Di Cerbo, I. Herwono, M. Manea, P. Mori, A. Sajjad, and X.-S. Wang, "A cloud-edge based data security architecture for sharing and analysing cyber threat information," *Future generation computer systems*, vol. 102, pp. 710–722, 2020.

[13] P. Liu, "Public-key encryption secure against related randomness attacks for improved end-to-end security of cloud/edge computing," *IEEE Access*, vol. 8, pp. 16 750–16 759, 2020.

[14] Y. Ren, G. Liu, V. Nitu, W. Shao, R. Kennedy, G. Parmer, T. Wood, and A. Tchana, "Fine grained isolation for scalable, dynamic, multi-tenant edge clouds," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, 2020, pp. 927–942.