Achieving Efficient Processing Time By Using IOT Smart Devices In Cloud-Based System

[1]**Kovuru Adithya Sai Kumar**, saikumar09401@gmail.com
[2]**M Naresh**, nareshmtech08@gmail.com
[1&2]Newton's Institute of Engineering, Guntur, Andhra Pradesh

Abstract
Over the last few years, smart devices are able to communicate with each other and with Internet/cloud from short to long range. As a consequence, a new paradigm is introduced called Internet of Things (IoT). However, by utilizing cloud computing, resource limited IoT smart devices can get various benefits like offload data storage and processing burden at cloud. To support latency sensitive, real-time data processing, mobility and high data rate IoT applications, working at the edge of the network offers more
benefits than cloud. In this paper, we propose an efficient data sharing scheme that allows smart devices to securely share data with others at the edge of cloud-assisted IoT. In addition, we also propose a secure searching scheme to search desired data within own/shared data on storage. Finally, we analyze the performance based on processing time of our proposed scheme. The results demonstrate that our scheme has potential to be effectively used in IoT applications.
Keywords: *Internet of Things (IoT), Cloud computing, Edge computing, Data sharing, Secure searching*

1. INTRODUCTION

The Internet of Things (IoT) connects billions of smart devices, transforming everyday environments into intelligent and autonomous systems across various sectors, including smart cities, healthcare, homes, and transportation. With Cisco predicting over 50 billion connected devices by 2020, the surge in IoT adoption has introduced vast data generation challenges, requiring efficient storage, processing, and secure data sharing. Although cloud computing offers virtually unlimited resources, it falls short in meeting the low latency, high data rate, and real-time processing demands of IoT applications. Edge computing addresses these gaps by processing data closer to devices, ensuring faster access, reduced latency, and efficient communication. Edge servers also act as intermediaries, strengthening connections between devices and cloud resources while enabling seamless data sharing. However, the decentralized nature of edge-based data sharing introduces significant security risks, including data leakage, unauthorized access, and integrity issues. Traditional security measures relying on resource-intensive cryptographic operations are impractical for resource-constrained IoT devices. To overcome these challenges, this paper proposes a lightweight cryptographic scheme that offloads security operations to edge servers, enabling secure data sharing among IoT devices. Additionally, the scheme includes a secure data-searching mechanism for encrypted data retrieval by authorized users. Performance analysis demonstrates the proposed solution's efficiency in reducing computational and communication overhead, paving the way for more secure and scalable cloud-assisted IoT environments.

2. LITERATURE SURVEY

The Internet of Things (IoT) promises to revolutionize daily life and business operations through seamless interactions among diverse devices. Despite decades of conceptual development, the fragmented landscape of communication technologies poses significant integration challenges, hindering the full realization of IoT's potential. The advent of 5G cellular systems offers a game-changing solution by providing ubiquitous, reliable, scalable, and cost-effective connectivity. Unlike previous cellular technologies primarily designed for broadband, 5G meets the unique requirements of IoT, such as massive device connectivity, lower energy consumption, and robust radio resource management. This paper explores the transformative potential of 5G for IoT by examining technological and standardization aspects. The current IoT connectivity landscape and highlight how 5G's capabilities can reshape business ecosystems, creating new opportunities for operators and vendors alike [R1]. In this paper, proposed the Edge-Fog Cloud, a decentralized model that efficiently distributes task processing across network resources. We introduce the Least Processing Cost First (LPCF) method for optimal task assignment, minimizing processing time while keeping network costs near optimal. Through comprehensive evaluations, we demonstrate the effectiveness of LPCF in various scenarios. The proposed model enhances data resilience via a centralized data store and reduces deployment time without increasing costs, offering a practical solution for real-world IoT applications [R2]. The Internet of Things (IoT) relies on numerous smart sensors that collect and share environmental data with cloud services for processing. To reduce the dependency on centralized cloud servers and minimize latency, architectural models like Fog and Edge computing have emerged, bringing processing closer to the data source. In this paper, we propose the **Edge-Fog Cloud**, a decentralized model that efficiently distributes task processing across network resources. We introduce the **Least Processing Cost First (LPCF)** method for optimal task assignment, minimizing processing time while keeping network costs near optimal. Through comprehensive evaluations, we demonstrate the effectiveness of LPCF in various scenarios. The proposed model enhances data resilience via a centralized data store and reduces deployment time without increasing costs, offering a practical solution for real-world IoT applications [R3]. The Internet of Things (IoT) goes beyond merely connecting countless endpoints — it introduces disruptive changes across computing and data processing landscapes. This chapter explores these disruptions and proposes **Fog Computing**, a hierarchical distributed architecture that extends from the network edge to the core. We highlight how Fog Computing complements and extends Cloud Computing by enabling service delivery through distributed compute, storage, and network resources. Key aspects of Fog's software architecture and its role in IoT and Big Data are discussed, along with use cases showcasing its relevance across various industries [R4]. The Internet of Things (IoT) is transforming connectivity across physical, cyber, and social spaces. However, its open data-sharing nature brings critical security challenges. Built on internet infrastructure, IoT inherits traditional vulnerabilities across its three layers: perception, transportation, and application. Addressing these security concerns is vital for secure and efficient operations. paper focuses on application layer security**,** highlighting key issues and technological solutions. It introduces end-to-end (E2E) security**,** embedding protection within the application payload with distinct confidentiality and authenticity trust domains. This approach supports capability-based access control and safeguards against eavesdropping. A comparative analysis of data security protection techniques for the application layer is also presented [R5]. Cloud computing is rapidly evolving due to its elastic, flexible, and on-demand storage and computing services. However, in cloud-based storage, data owners relinquish control to third-party cloud service providers (CSPs),

raising significant security concerns, especially when sharing data with other users. To address these challenges, various cryptographic schemes have been proposed. In this paper, we present a secure data-sharing model that ensures **data confidentiality, access control**, and **dynamic user management**. The proposed model eliminates the need for users to manage encryption keys or be online for data access requests. It also supports dynamic user membership changes, offering a scalable and secure solution for cloud environments [R6]. Later Proposed a **mediated certificate-less public key encryption (mCL-PKE) scheme without pairing operations** for secure data sharing in public clouds. Traditional mCL-PKE schemes either face inefficiencies due to computationally expensive pairing operations or are vulnerable to partial decryption attacks. proposed mCL-PKE scheme addresses these challenges by eliminating pairing operations and providing a practical solution for secure information sharing. The cloud acts as both a secure storage platform and key generation station. Data owners encrypt sensitive data using cloud-generated public keys based on access control policies and upload the encrypted data to the cloud. Upon successful authorization, the cloud performs partial decryption, enabling authorized users to fully decrypt the data using their private keys. This approach solves key escrow and retrieval issues while enhancing efficiency in multi-user environments. Experimental results demonstrate that our scheme reduces encryption overhead for data owners and ensures secure and efficient data sharing in public cloud environments [R7]. Cloud-integrated Internet of Things (IoT) is emerging as a next-generation service platform, driving smart applications such as smart grids, e-health systems, and large-scale environmental monitoring. These applications generate vast amounts of data, requiring cloud services for efficient storage and analysis. However, secure and privacy-preserving implementations are essential to protect data integrity and prevent unauthorized access. This paper explores the security challenges in enabling cloud-based data analytics for IoT. We discuss three key application areas and propose solutions using fully homomorphism encryption (FHE) to safeguard data during cloud-based analytical processes. Existing limitations are examined, and efficient models are proposed to achieve secure and accurate analytics for encrypted data. Moreover, we present a survey on IoT and cloud computing, highlighting their convergence and security challenges. By examining their common features and trade-offs, we demonstrate how integrating these technologies can unlock robust and secure solutions for emerging IoT applications [R8]. The Internet of Things (IoT) connects numerous agents for diverse applications, requiring seamless collaboration through predefined protocols. In this paper, we propose a general trust management framework to evaluate the trustworthiness of agents in IoT systems. The framework employs two key metrics: trustworthiness (m) and confidence (c)**,** based on measurement theory. Agents assess each other's trust levels through various environments and factors. To demonstrate its effectiveness, we run a food nutrition analysis for diabetes treatment**,** showing how trust-based evaluations reduce analysis errors and filter inaccurate information. We also highlight two potential attack scenarios and illustrate how selecting and weighting trust factors can mitigate damage, emphasizing the importance of adaptive trust management in IoT systems [R9]. Cyber-Physical Systems (CPS) are mission-critical systems combining cyber and physical components, often constrained by limited resources and real-time demands. To enhance their efficiency, CPS are integrated with cloud computing, forming Cyber-Physical Cloud Computing Systems (CPCCS)**,** which are increasingly used in critical care applications where security is paramount. This paper proposes a secure service provisioning architecture for CPCCS, integrating CPS, Cloud Computing, and Wireless Sensor Networks. We

highlight various security challenges, threats, and mechanisms applicable at different system layers and introduce two security models Horizontal and Vertical tailored for CPCCS [R10].

3. MATERIALS AND METHODS

In this paper, by considering the aforementioned limitations of current solutions for resource- limited smart devices, we propose a lightweight cryptographic scheme so that IoT smart devices can share data with others at the edge of cloud-assisted IoT wherein all security-oriented operations are offloaded to nearby edge servers. Furthermore, although initially we focus on data-sharing security, we also propose a data-searching scheme to search desired data/shared data by authorized users on storage where all data are in encrypted form.

3.1 Process Model

DLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software. The requirements gathering process begins with the goals outlined in the project's high-level requirements section. These goals are refined into specific requirements that define the key functions of the application, including operational and reference data areas, as well as initial data entities. Major functions encompass critical processes, inputs, outputs, and reports. A user class hierarchy is created and linked to these functions, data areas, and entities. Each definition is a Requirement, identified by a unique ID and, at a minimum, includes a title and a textual description.

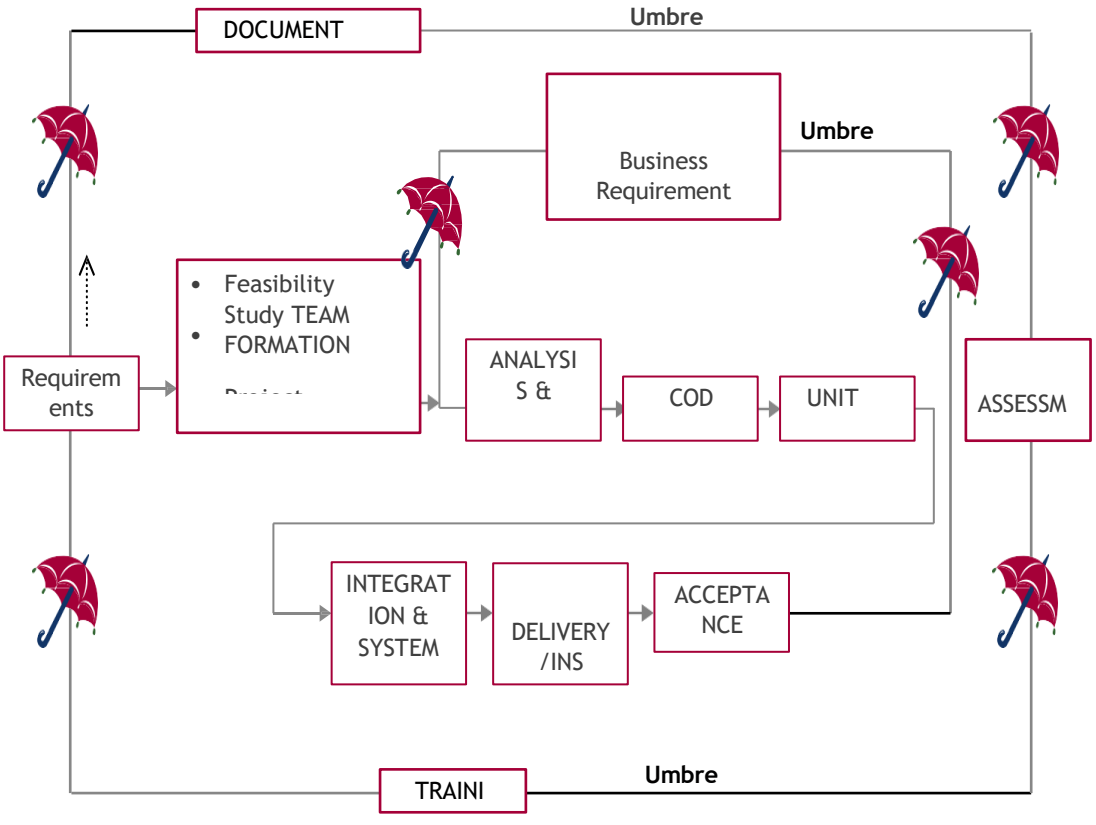

Fig.1. Proposed Model

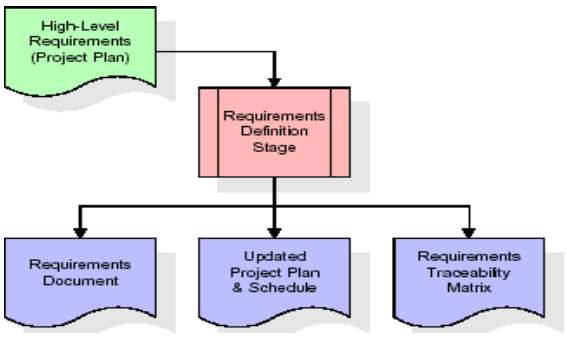

Fig.1. Requirements Gathering

The Requirements Document and Requirements Traceability Matrix (RTM) are the key deliverables in the requirements stage. The document details each requirement, with diagrams and references but excludes database specifics. The RTM links each requirement to its corresponding project goal, ensuring traceability across the development stages. Key outputs include the requirements document, RTM, and updated project plan. Feasibility study identifies project issues, Team formation defines staff roles and task assignments, and Project specifications outline inputs, outputs, and admin-managed reports.

Analysis Stage: The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

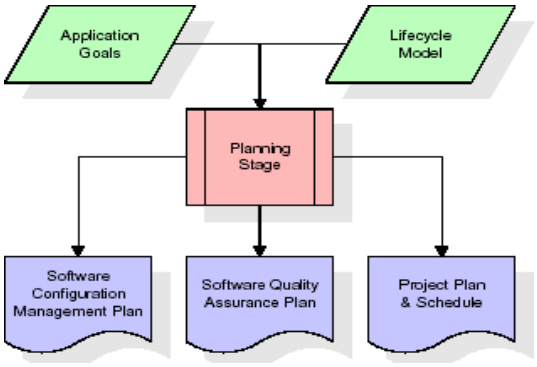

Fig.3. Analysis Stage

The project plan's most crucial section outlines high-level product requirements, or goals, which guide the development of software requirements during the definition stage. Each goal includes a title and description, with optional references. Outputs of the project planning stage include the configuration management plan**,** quality assurance plan**,** project plan**,** and schedule**,** detailing activities for the Requirements stage and effort estimates for future stages.

Designing Stage: The project plan's most crucial section outlines high-level product requirements, or goals, which guide the development of software requirements during the definition stage. Each goal includes a title and description, with optional references. Outputs of the project planning stage include the configuration

management plan**,** quality assurance plan**,** project plan**, and** schedule**,** detailing activities for the Requirements stage and effort estimates for future stages.

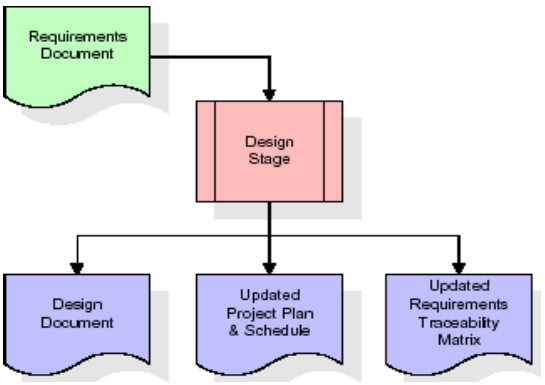

Fig.4. Design Stage

Development (Coding) Stage: The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software. The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

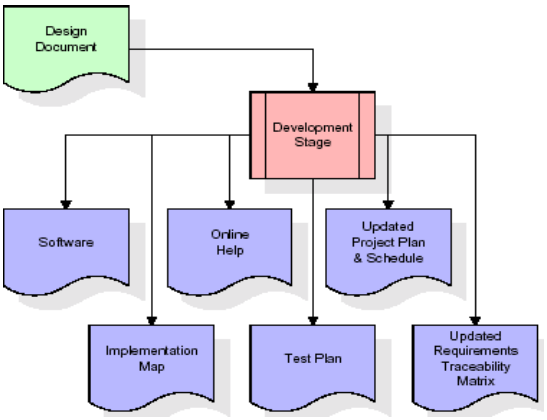

Fig.5. Development Stage

Installation & Acceptance Test: During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to

verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer. After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software. The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

Implementation and Testing: Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

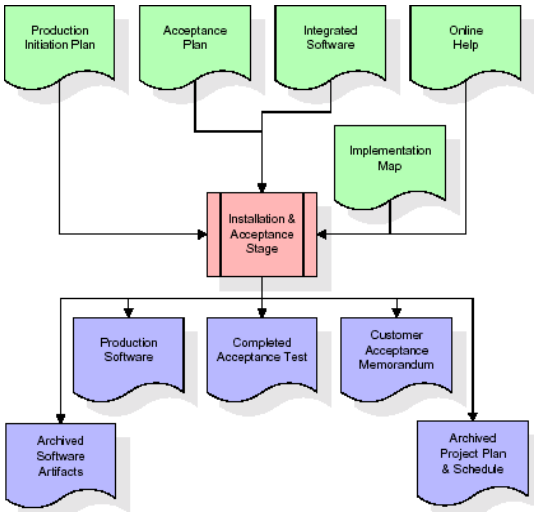

Fig.6. Implementation and Testing Stage

4. RESULT

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Run cloud storage server	Verify whether server is running or not	If not server starts	The cloud server services are not started	The cloud server services are started homepage displayed	Low	Medium

02	Run Edge server	Verify whether server is running or not	If not server starts	The Edge server services are not started	The Edge server services are started	High	High
03	Run Key generator	Verify whether key generator running or not	If not server starts	The Key generator server services are not started	The Key generator server services are started	Low	High

Table 1: Comparison models

5. CONCLUSION

In this paper, we present a proposed data-sharing and -searching scheme to share and search data securely by IoT smart devices at the edge of cloud-assisted IoT. The performance analysis demonstrates that our scheme can achieve better efficiency in terms of processing time compared with existing cloud-based systems. In future work, we plan on authenticating and accessing control challenges in this area. We hope that our proposed scheme is practical to be deployed and opens a new door in edge-oriented security research for cloud assisted IoT applications.

REFERENCES

[1] M.R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, et al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," IEEE J. Selected Areas in Communications, vol. 34, no. 3, 2016, pp. 510–527.

[2] 2.L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds," IEEE Cloud Computing, vol. 2, no. 1, 2015, pp. 76–80.

[3] 3.M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, et al., "Edge Analytics in the Internet of Things," IEEE Pervasive Computing, vol. 14, 2015, pp. 24–31.

[4] 4.S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and Applications," 2015 3rd IEEE Workshop Hot Topics Web Systems and Technologies (HotWeb), 2015, pp. 73–78.

[5] 5.J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty Security Considerations for CloudSupported Internet of Things," IEEE Internet of Things J., vol. 3, no. 3, 2016, pp. 269– 284.

[6] 6.M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A.V. Vasilakos, K. Li, et al., "SeDaSC: Secure Data Sharing in Clouds," IEEE Systems J., vol. 99, 2015, pp. 1–10.

[7] 7.S.-H. Seo, M. Nabeel, X. Ding, and E. Bertino, "An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds," IEEE Trans. Knowledge and Data Engineering, vol. 26, no. 9, 2014, pp. 2107–

2119.

[8] 8.H. Kumarage, I. Khalil, A. Alabdulatif, Z. Tari, and X. Yi, "Secure Data Analytics for CloudIntegrated Internet of Things Applications," IEEE Cloud Computing, vol. 3, no. 2, 2016, pp. 46–56.

[9] 9.J.B. Bernabe, J.L.H. Ramos, and A.F.S. Gomez, "TACIoT: Multidimensional Trust-Aware Access Control System for the Internet of Things," Soft Computing, vol. 20, no. 5, 2016, pp. 1763–1779.

[10] 10.F. Li, Y. Rahulamathavan, M. Conti, and M. Rajarajan, "Robust Access Control Frameworfor Mobile Cloud Computing Network," Computer Communications, vol. 68, 2015, pp. 61–72.