Bridging Security and Efficiency: Energy-Aware Clustering for Data Encryption in Smart Agriculture

Huzaif Khan*, Lokesh Purohit†, Ali Jalooli‡
Department of Computer Science, California State University Dominguez Hills, Carson, CA 90747, USA
Emails: hkhan17@toromail.csudh.edu*, lpurohit1@toromail.csudh.edu†, ajalooli@csudh.edu‡

***Abstract*—Securing data communication in smart agriculture is essential to ensure data integrity and confidentiality in distributed sensor networks. While Cheon-Kim-Kim-Song (CKKS) homomorphic encryption offers strong security by allowing computation on encrypted data, its significant computational and energy overhead makes it unsuitable for energy-constrained devices like those in agricultural monitoring systems. We propose a novel framework to enable CKKS encryption in resource-limited settings through an optimized, energy-efficient clustering approach. By reducing the number of direct transmissions through data aggregation at cluster heads, our approach mitigates CKKS's energy demands, making it feasible for low-power sensors. We formulate the clustering problem as an NP-hard optimization problem and present a real-time clustering algorithm that minimizes energy consumption while adapting to dynamic sensor energy levels. To evaluate our approach, we use the Gurobi optimizer to solve the problem to optimality and benchmark our algorithm's performance. Results show that our solution achieves near-optimal energy efficiency, closely approximating the optimal solution while remaining computationally feasible for real-time applications. Additionally, our method improves packet delivery ratios and reduces end-to-end delay. This work provides a pathway for the practical implementation of advanced encryption schemes in energy-constrained environments.**

***Index Terms*—CKKS, Optimal Clustering, Data Aggregation, Sensor Network.**

I. Introduction

In the era of smart agriculture, fields are increasingly monitored by advanced sensors and drones, optimizing water usage, crop health, and farm management in real time. This data-driven approach enables automated operations and tailored farming practices, crucial for global food security and addressing the impending food crisis. With the global population expected to reach 10 billion by 2050, transitioning to smart agriculture is essential for sustaining a growing world [1]. The benefits are transformative: automation enhances operational efficiency, reduces labor, and increases precision, while resource optimization minimizes waste and environmental impact. Data analytics and artificial intelligence improve crop yields and food production capabilities, advancing sustainability through reduced carbon footprints and chemical use. Real-time monitoring empowers farmers to make informed decisions, improving supply chain transparency, financial efficiency, climate adaptability, and labor management. Despite these advancements, integrating digital technologies introduces significant security and privacy challenges. The extensive networks of interconnected devices that underpin smart agriculture increase vulnerability to cyber threats [2]. The valuable data generated and utilized in innovative farming operations become potential targets for malicious attacks, threatening the integrity and safety of food production systems. Integrating diverse technologies such as mobile devices, precision agriculture tools, remote sensing, and cloud analytics increases the complexity and vulnerability of these systems. This is especially true in open-field agriculture, where devices are exposed to external elements and cyber threats. Issues such as compatibility, heterogeneity, and the need for mass data protection further complicate security, rendering traditional security measures less effective.

To address security concerns in smart agriculture, Cheon-Kim-Kim-Song (CKKS) homomorphic encryption [3] has emerged as a promising solution. CKKS enables computations on encrypted data, ensuring confidentiality throughout processing and transmission. This is particularly valuable in smart agriculture, where sensitive data such as soil conditions and crop health must remain secure across distributed sensor networks. However, CKKS's high computational and energy demands pose a significant challenge for battery-powered sensors in agricultural environments, leading to rapid battery depletion and hindering practical deployment. This work aims to make CKKS encryption feasible in energy-constrained agricultural settings by addressing its high energy requirements through an adaptive clustering framework. By organizing sensors into clusters and minimizing direct transmissions via data aggregation at cluster heads (CHs), our solution significantly reduces the energy overhead of CKKS encryption. This approach ensures that CKKS becomes a viable option for real-time, secure communication in agricultural setups, enabling secure and energy-efficient data transmission in resource-constrained environments. The main contributions of this paper are as follows:

- We formulate the clustering problem as an energy optimization challenge for sensor networks using CKKS encryption. By proving the problem is NP-hard, we highlight the complexities of optimizing energy efficiency in secure data transmission, providing theoretical insight into the trade-offs between encryption, energy use, and network performance.
- We propose an adaptive clustering algorithm that adjusts to fluctuating sensor energy levels through periodic re-optimization. This ensures continuous adaptation to changing conditions, maintaining security and energy

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

efficiency, and directly addressing the energy demands of CKKS encryption.
- We validate our approach through simulations, benchmarking it against alternative algorithms and the optimal Gurobi solution [4]. Our results demonstrate near-optimal performance with significant improvements in energy efficiency, packet delivery ratios, and network reliability, especially in energy-constrained environments.

The paper is organized as follows: Section II reviews related work, while Section III presents the system model and problem formulation for energy-efficient sensor clustering. Section IV introduces the Energy-Aware CKKS Clustering (ECKC) approach and the EnergyGreedy Clustering Algorithm. Section V evaluates the system's performance, and conclusions are drawn in Section VI.

II. Related Works

This section reviews recent advancements in data security and efficiency in smart agriculture, focusing on encryption techniques and data management in IoT-enabled environments. Zhou et al. [5] proposed the K2-treap scheme, integrating BGN homomorphic encryption to protect data on IoT-enabled fog nodes. While this approach enhances the efficiency of handling and updating range queries while preserving data privacy, its applicability is limited by the substantial computational overhead and energy requirements associated with homomorphic encryption. Singhal and Singh [6] implemented Scalable Bloom Filters to optimize data storage and transmission in smart agriculture. This method effectively balances data redundancy and security but does not address the energy consumption issues inherent in sensor networks, which are critical for maintaining operational efficiency. Nguyen et al. [7] introduced the Correlated Low-rank Structure (CoLR), which effectively reduces communication overheads in federated learning systems, enhancing privacy while maintaining performance. However, this approach does not adequately address the energy efficiency challenges inherent to sensor networks or the computational burden of processing large-scale data in real-time scenarios. Vangala et al. [8] outlined a multi-layered security framework for IoT-based farm ecosystems, which provides robust data integrity. Despite this, the framework may be challenged by the increased complexity and resource demands of implementing such a comprehensive defense strategy across extensive sensor networks. Taji and Ghanimi [9] explored hyper-elliptic curve cryptography for smart agriculture, aiming to resolve computational bottlenecks in encryption. Although this method mitigates some computational issues, it does not adequately address the practical energy constraints of battery-powered sensors or the dynamic nature of agricultural environments. Several studies have also aimed to enhance security in smart agriculture by leveraging lightweight cryptography technology. Works such as [10] have explored dynamic cryptographic mechanisms to optimize energy consumption in IoT environments. Their approach introduced adaptive encryption parameters tailored to available power resources and communication distances, significantly improving power efficiency and node longevity. Another study [11] proposed a proxy re-encryption (PRE) scheme tailored for WSNs, combining lightweight symmetric and asymmetric cryptographic techniques. Their approach minimized computational costs, enhanced secure key management, and extended network lifetime while maintaining robust security and efficiency. In [12], a lightweight key distribution mechanism for WSNs was proposed, integrating optimal cluster head selection, improved ECC-based encryption, and efficient key management. Their approach utilized a hybrid optimization algorithm for clustering and secure data transmission, enhancing scalability, security, and energy efficiency.

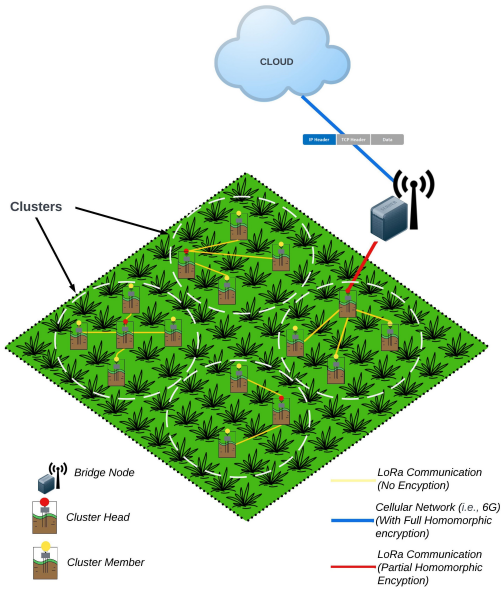

Fig. 1. System Model

III. System Model

In this section, we outline the network model utilized in this study, followed by the formulation of the energy-aware clustering problem.

A. Smart Agriculture Network

Figure 1 depicts the system model used in this study. Let $\mathbf{S} = \{\mathbf{s_0}, \ldots, \mathbf{s_{|S|-1}}\}$ denote a set of IoT sensors distributed across a large field of crops. These multi-functional sensors can collect various environmental data, including soil moisture, temperature, toxicity, etc. Initially, sensors are divided into random clusters based on their geographical locations. The optimal clustering algorithm will be discussed in Section IV. Each cluster has a CH that regulates channel utilization, performs data aggregation, and manages internal and inter-cluster signaling. Each cluster also includes a set of Cluster Members (CMs). Each sensor maintains a table with a unique Sensor ID, battery percentage, and a head flag indicating its role as a CH or CM.

The network also includes a set of bridge nodes $\mathbf{BN} = \{\mathbf{bn_0}, \ldots, \mathbf{bn_{|BN|-1}}\}$. The bridge node is responsible

for calculating the optimal set of CHs and their corresponding CMs, aggregating data from CHs, handling the main types of packets that the network supports (i.e., clustering and data packets), performing homomorphic encryption on the data packets, and forwarding the encrypted data to the Cloud server. Clustering packets are used for managing and coordinating the clustering process, while data packets contain the environmental data collected by the sensors. The cloud server is the central repository and analytical engine, storing data from the bridge node and performing advanced computations to generate insights and optimize farm management. The sensors and bridge nodes communicate using LoRa, a low-power, wide-area network (LPWAN) technology for long-range connectivity.

B. Problem Formulation

Let $D_{i,j}$ denote the distance between sensors i and j, and let D_{max} represent the maximum allowable distance for effective communication between nodes. Let B_i denote the battery level of the node i, and B_{min} be the minimum battery threshold required for a node to be eligible to serve as a CH. Additionally, let CM_{max} indicate the maximum number of nodes that a CH can effectively manage. We also define the energy consumption associated with transmitting and processing data. Let $E_{\text{trans}}(i,j)$ represent the energy consumed by node i to transmit data to CH j, and $E_{\text{comp}}(j)$ represent the energy required by CH j to process the data it receives. The total energy consumption of a CH j is

$$E_{\text{CH}}(j) = \sum_{i \in \mathbf{S}} C_{i,j} E_{\text{trans}}(i,j) + E_{\text{comp}}(j) \tag{1}$$

For each node i, the total energy consumption E_i is defined as

$$E_i = \begin{cases} E_{\text{CH}}(j) & \text{if node } j \text{ is a CH} \\ E_{\text{trans}}(i,j) & \text{if node } i \text{ is connected to CH } j \end{cases} \tag{2}$$

In the following, We formally define the problem as an optimization that aims to minimize the number of CHs and the total energy consumption in the network. A weighting factor α is also introduced to balance the trade-off between minimizing the number of CHs and energy consumption.

$$CH_i = \begin{cases} 1 & \text{if node } i \text{ is selected as a CH} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$C_{i,j} = \begin{cases} 1 & \text{if node } i \text{ is connected to CH}j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\text{Min} \sum_{i \in \mathbf{S}} CH_i + \alpha \sum_{i \in \mathbf{S}} E_i \tag{5}$$

$$\sum_{j \in \mathbf{S}} C_{i,j} = 1 \quad \forall i \in \mathbf{S} \tag{6}$$

$$C_{i,j} \leq CH_j \quad \forall i,j \in \mathbf{S} \tag{7}$$

$$CH_i = 0 \quad \text{if } B_i < B_{\text{min}} \quad \forall i \in \mathbf{S} \tag{8}$$

$$C_{i,j} = 0 \quad \text{if } D_{i,j} > D_{\text{max}} \quad \forall i,j \in \mathbf{S} \tag{9}$$

$$\sum_{i \in \mathbf{S}} C_{i,j} \leq CM_{\text{max}} \quad \forall j \in \mathbf{S} \tag{10}$$

$$C_{j,k} \leq 1 - CH_j \quad \forall j \neq k \in \mathbf{S} \tag{11}$$

Constraint (6) ensures that each node must be connected to exactly one CH. Constraint (7) stipulates that a node can only be connected to a CH if that CH is selected. Constraint (8) specifies that only nodes with battery levels above a minimum threshold can be CHs. Constraint (9) restricts connections between nodes based on distance. Constraint (10) limits the number of nodes that a CH can manage, preventing any single CH from being overloaded and ensuring balanced clustering. Finally, Constraint (11) prevents CHs from being part of any other cluster.

Theorem 1. The Dynamic Cluster Head Selection (DCHS) problem in Wireless Sensor Networks (WSNs) is NP-hard.

Proof. To establish the NP-hardness of the DCHS problem, we reduce it to the Set Cover problem, a well-known NP-hard problem. In the Set Cover problem, given a universe U and a family of subsets $S = \{S_1, S_2, \ldots, S_m\}$ where each subset $S_i \subseteq U$ has an associated cost $c(S_i)$, the goal is to select a subfamily $R \subseteq S$ such that $\bigcup_{S_i \in R} S_i = U$ and the total cost $\sum_{S_i \in R} c(S_i)$ is minimized. We map the DCHS problem to the Set Cover problem by defining the universe U as the set of all sensor nodes and each subset S_j as the set of all nodes within the communication range D_{max} of a potential CH CH_j. The cost $c(S_j)$ is set to $\frac{1}{B_j}$, where B_j is the battery level of CH_j, reflecting a preference for CHs with higher battery levels. The DCHS problem seeks to minimize the number of CHs while covering all sensor nodes and minimizing the total cost, aligning directly with the Set Cover problem's objective. Since the reduction from DCHS to Set Cover involves polynomial-time operations, solving DCHS efficiently would imply solving Set Cover efficiently, proving that DCHS is NP-hard. ■

IV. Energy-Aware CKKS Clustering (ECKC)

As established, DCHS problem is NP-hard, making the identification of optimal clusters computationally intractable, particularly in real-time applications for sensor networks. This issue is exacerbated by the dual constraints of limited battery life and the significant energy demands imposed by CKKS encryption.To address these challenges, we present the Energy Greedy Clustering Algorithm (EGCA), the central element of our ECKC framework, designed to balance computational feasibility and energy conservation. EGCA operates using a real-time periodic optimization approach executed by the bridge node, which allows it to periodically generate and adapt clusters based on real-time fluctuations in sensor energy

levels. This approach ensures that the clustering remains both efficient and responsive to dynamic network conditions. In the following subsections, we provide a detailed introduction to EGCA, explaining how it constructs resilient clusters and optimizes energy usage while maintaining the practicality of CKKS encryption.

A. EGCA

The EGCA optimizes clustering in Wireless Sensor Networks (WSNs) by reducing the number of CHs, which in turn lowers energy consumption and enables the use of energy-intensive algorithms such as CKKS. EGCA utilizes a K-D Tree structure to manage spatial relationships among nodes more efficiently, improving computational performance. At the start, the algorithm initializes empty sets for assigned nodes ($\mathcal{A}$) and clusters ($\mathcal{C}$) and constructs a neighbor list ($\mathcal{L}$) using a K-D Tree ($\mathcal{T}$) based on the node set $\mathcal{N}$. This structure allows for efficient nearest-neighbor searches, reducing the time complexity of neighbor identification from $O(n)$ to $O(\log n)$ per query, where n represents the number of nodes. For each node i, EGCA identifies neighbors within a communication range $D_{\max}$, storing them in $\mathcal{L}[i]$. Nodes with battery levels $B_i \leq B_{min}$ are classified as critical due to limited energy reserves. The algorithm then compiles a list called counts, which includes tuples for each node i containing the count of critical neighbors c, total neighbors $|\mathcal{L}[i]|$, and battery level $\mathbf{B}[i]$. This list is sorted in descending order, prioritizing nodes with many critical neighbors, extensive connectivity, and higher energy levels. The algorithm processes the sorted list, assigning unallocated nodes with battery levels above B_{min} as CHs, adding them to $\mathcal{C}$. Each cluster head incorporates unassigned neighbors into its cluster, up to a maximum of $CM_{\max}$ members, marking them as assigned in $\mathcal{A}$. The K-D Tree expedites this process, keeping neighbor retrieval and cluster assignment at an average time complexity of $O(\log n)$. If any nodes remain unassigned after initial clustering, they are either assigned to the nearest available cluster or become new CH to ensure complete network coverage with a minimal number of CHs.

The achievements of EGCA include significant energy efficiency by selecting high-energy nodes as CHs and minimizing their number, ensuring reduced energy consumption in managing clusters. The K-D Tree's integration enhances scalability, making the algorithm suitable for large-scale WSNs. Additionally, EGCA supports critical nodes by prioritizing those with many low-energy neighbors and facilitates the execution of energy-intensive algorithms such as CKKS. Furthermore, by reducing the number of CHs, the algorithm minimizes potential failure points, increasing the network's security. EGCA provides an efficient, energy-aware clustering solution for WSNs, enhancing sustainability and security while supporting advanced cryptographic operations.

Algorithm 1: EGCA

Input: $\mathbf{S}$, $\mathbf{B}$, $D_{\max}$, $CM_{\max}$, $B_{\min}$
Output: $\mathcal{C}$
Initialize $\mathcal{A} \leftarrow \emptyset$, $\mathcal{C} \leftarrow \emptyset$, $\mathcal{L} \leftarrow$ defaultdict(list);
Construct KD-Tree $\mathcal{T}$ from $\mathbf{S}$;
foreach $i \in \{0, \ldots, |\mathbf{S}| - 1\}$ **do**
  Find neighbors $\mathcal{L}[i] \leftarrow \{j \mid j \in$ neighbors of $i, j \neq i, d(i,j) \leq D_{\max}\}$;
  if $\mathbf{B}[i] \leq B_{min}$ **then**
    Add i to critical nodes;
Initialize counts $\leftarrow []$;
for $i \in \{0, \ldots, |\mathbf{S}| - 1\}$ **do**
  $c \leftarrow |\{j \in \mathcal{L}[i] \mid j \in$ critical nodes$\}|$;
  Append $(i, c, |\mathcal{L}[i]|, \mathbf{B}[i])$ to counts;
Sort counts by decreasing c, $|\mathcal{L}[i]|$, $\mathbf{B}[i]$;
foreach $(i, _, _, _) \in$ *counts* **do**
  if $i \in \mathcal{A}$ ***or*** $\mathbf{B}[i] \leq B_{min}$ **then**
    continue;
  Set i as cluster head, add to $\mathcal{C}$;
  Initialize $\mathcal{M} \leftarrow []$;
  Add i to $\mathcal{A}$;
  foreach $j \in \mathcal{L}[i]$ **do**
    if $|\mathcal{M}| \geq CM_{max}$ **then**
      break;
    if $j \notin \mathcal{A}$ **then**
      Add j to $\mathcal{M}$, $\mathcal{A}$;
  Add $(i, \mathcal{M})$ to $\mathcal{C}$;
unassigned $\leftarrow \mathbf{S} \setminus \mathcal{A}$;
foreach $i \in$ *unassigned* **do**
  Find nearest cluster with capacity for i;
  if *exists* **then**
    Assign i to nearest cluster, add to $\mathcal{A}$;
  else
    Set i as new cluster head, add to $\mathcal{C}$, $\mathcal{A}$;
return $\mathcal{C}$

B. Energy-Aware Clustering for Efficient CKKS Encryption

Our proposed architecture supports two key packet types: clustering packets and data packets. In this section, we first elaborate on the flow of clustering packets, which are critical for dynamically forming optimal clusters, and then describe the flow of data packets, detailing how energy-efficient encryption is achieved through our clustering mechanism.

1) Clustering Packets: At the outset, sensors are randomly grouped into clusters. Each CM sends periodic *hello packets* to its CH at intervals denoted by C_t, where C_t is the time between two successive transmissions from a CM. The *hello packet* contains critical information such as the node's identifier (ID), current battery percentage, and a timestamp T_{hello}. Each CH aggregates these packets over a duration C_T, such that $C_t < C_T$, ensuring the collection of up-to-date network information. Let $H_{CM}(t)$ represent the set of hello packets

received by a CH at time t, which can be defined as:

$$H_{CM}(t) = \{(ID_i, B_i, T_{hello,i}) \mid i = 1, \ldots, N_{CM}\}$$

where ID_i is the identifier, B_i is the battery percentage, and $T_{hello,i}$ is the timestamp of sensor i. After the CHs accumulate these packets, they forward the information to the bridge node. At regular intervals, the bridge node executes the EGCA, which solves the clustering optimization problem by finding the configuration that minimizes the number of CHs and energy consumption. Once the bridge node computes the cluster assignments, the results are multicast to all CHs. Each CH, in turn, disseminates the updated cluster information to its CMs. If a CM's assignment has changed (i.e., from CM to CH or vice versa), its role is updated accordingly. These assignments remain in effect until the next scheduled optimization, during which the algorithm reevaluates and updates cluster configurations. Until that time, all sensor nodes maintain their current roles as either CM or CH to ensure stability between updates while allowing the system to adapt to dynamic environmental and network conditions.

In addition to improving energy efficiency, our periodic re-optimization framework enhances security by reducing the exposure of critical nodes. Since the CH roles are reassigned at regular intervals, the time window during which an attacker could exploit a CH is limited, thus mitigating the risk of long-term attacks. Furthermore, by regularly updating cluster assignments, the system disrupts static communication patterns, making it more difficult for adversaries to perform traffic analysis and identify key nodes for targeted attacks. This dynamic reassignment of roles also reduces the impact of compromised nodes, as a node acting as a CH in one cycle may be reassigned as a CM in the next cycle, significantly lowering its control over the network. Additionally, the periodic reevaluation allows the system to demote potentially suspicious nodes from CH status, reducing the risk of sustained attacks from compromised nodes. Therefore, by continuously adapting cluster roles in response to changing environmental and network conditions, the system ensures both security and operational robustness, maintaining resilience against evolving threats

2) Data Packets: Each sensor node periodically collects environmental data and transmits it to its designated CH at intervals D_t, where D_t represents the data transmission interval. The CH aggregates the data packets over a window D_T, where $D_t < D_T$. The aggregated data at the CH can be represented as:

$$D_{agg}(t) = \sum_{i=1}^{N_{CM}} D_i(t) \tag{12}$$

where $D_i(t)$ is the data from CM i at time t. To handle data redundancy, the CH applies a hashing function $H : \mathbb{P} \to H(\mathbb{P})$, where $\mathbb{P}$ is the set of all received data packets. The hash function computes a unique key $\mathcal{K}$ for each data packet based on its ID and timestamp:

$$\mathcal{K} = h(\text{ID}, T_{data})$$

where T_{data} is the timestamp of the data packet. Redundant packets are identified and removed by checking for collisions in the hash table $H(\mathbb{P})$, with collision resolution performed using separate chaining. After aggregation, the CH applies partial encryption to the data to reduce computational overhead while still ensuring security. The partially encrypted data is transmitted to the bridge node, where final encryption using the CKKS homomorphic encryption scheme is performed. In CKKS, each plaintext message m is represented as a polynomial $P(x)$, such that:

$$m \to P(x) = a_0 + a_1 x + \cdots + a_n x^n \tag{13}$$

The coefficients $\{a_0, a_1, \ldots, a_n\}$ encode the message, which is encrypted using a public key pk, resulting in the ciphertext:

$$\text{Enc}_{pk}(P(x)) \tag{14}$$

CKKS allows operations on the encrypted polynomials directly, such that the sum of two ciphertexts $\text{Enc}_{pk}(P(x))$ and $\text{Enc}_{pk}(Q(x))$ can be computed as:

$$\text{Enc}_{pk}(P(x) + Q(x)) = \text{Enc}_{pk}(R(x)) \tag{15}$$

where $R(x)$ is the resulting polynomial. The ciphertext is then transmitted to the cloud server, where it is decrypted using the private key sk, producing an approximation of the original message. This two-stage encryption process ensures secure transmission while managing the energy demands of CKKS encryption.

V. Performance Evaluation

A. Simulation Setup

We simulated a 10-hectare agricultural field over a 24-hour period, capturing normal operation, increased traffic, and low battery conditions to evaluate the robustness of our clustering algorithm. Using the SimPy framework, we modeled sensor deployments ranging from 20 to 120 nodes, uniformly distributed across the field. LoRa technology was used for its long-range communication and energy efficiency. Key parameters included a maximum of 10 CMs per CH, clustering decisions updated every 10 minutes, data packets sent every 5 minutes, and data aggregation performed every 30 minutes. Data packets were 1024 bytes, and clustering packets were 128 bytes.

B. Simulation Results

To evaluate the effectiveness of our proposed EGCA approach, we use the Gurobi optimizer to solve the optimization problem (Eq. 5) to optimality, providing a benchmark for our algorithm's performance. Additionally, we compare EGCA with the LEACH algorithm [13], a widely used clustering method in sensor networks. In the following, we first examine the impact of clustering algorithms on network performance, focusing on key metrics such as Packet Delivery Ratio (PDR), which represents the ratio of successfully delivered packets to the total number generated, and End-to-End Delay, which measures the time required for a packet to travel from a sensor

to the cloud server. We then assess the effect of clustering on energy consumption and its significance in enabling energy-intensive encryption techniques, such as CKKS. Fig. 2 depicts the PDR for four different configurations. The No Clustering configuration, which refers to a scenario where all sensor nodes communicate directly with the bridge node without forming clusters, shows the lowest performance, with PDR ranging from 50% to 70% due to congestion and inefficient communication. However, as the number of nodes increases, the PDR improves slightly due to increased node density, allowing more reliable communication. This underscores the need for clustering to manage traffic and improve efficiency. LEACH shows moderate improvement, with PDR increasing from approximately 70% to 80%, but its random CH selection results in suboptimal performance. In contrast, EGCA achieves near-optimal performance, with PDR consistently above 95%, closely matching the PDR of the Optimal Solution.

Fig. 3 presents the comparison of End-to-End Delay across four configurations. The No Clustering configuration demonstrates the highest delay, with a consistent increase from approximately 8 ms to nearly 16 ms as the number of nodes increases. LEACH follows a similar upward trend, though with slightly lower delays. In contrast, EGCA achieves significantly lower delays, starting at approximately 3.5 ms and increasing only slightly as the network grows, demonstrating superior scalability. Notably, EGCA's performance closely approaches that of the Optimal Solution, which maintains the lowest delay, ranging from just over 2 ms to around 3 ms. This highlights EGCA's effectiveness in delivering near-optimal delay performance while balancing computational efficiency.

Next, we examine the impact of clustering algorithms on energy consumption, specifically focusing on how efficient clustering can mitigate the energy drain on sensor nodes. This analysis is critical for understanding how different clustering strategies can support the use of energy-intensive homomorphic encryption schemes, such as CKKS, to enhance data security while maintaining battery longevity and network performance. Fig. 4 demonstrates the substantial energy savings achieved through efficient clustering mechanisms. The No Clustering, No CKKS configuration yields the lowest energy consumption, as no encryption is applied, resulting in minimal computational demand. In contrast, the CKKS with No Clustering configuration incurs the highest energy consumption due to poor network performance and the additional communication overhead imposed on each sensor node. LEACH, which is designed to prioritize energy efficiency, shows energy consumption levels similar to those of EGCA. However, as shown in Figs. 2 and 3, LEACH's network performance is significantly compromised, leading to higher delays and lower packet delivery ratios. The EGCA algorithm, by contrast, effectively balances energy efficiency and the computational overhead introduced by CKKS encryption, making it a practical and viable solution for secure, energy-constrained IoT environments.

Fig. 5 presents the execution time required for Gurobi, EGCA, and LEACH to compute clusters as the network size

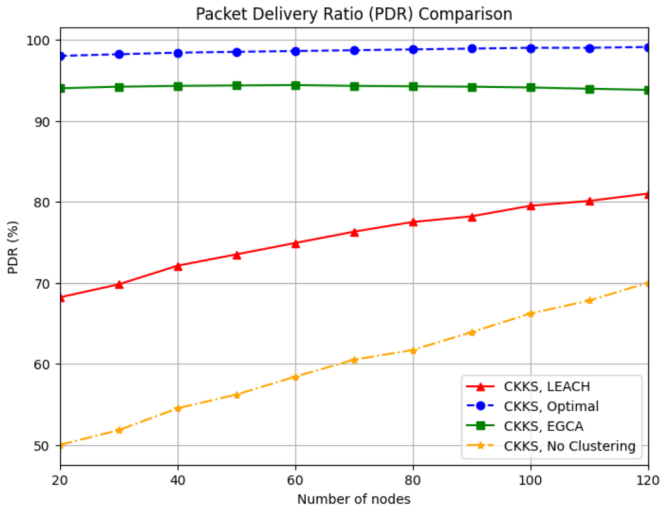

Fig. 2. PDR as a function of the number of nodes for different clustering configurations

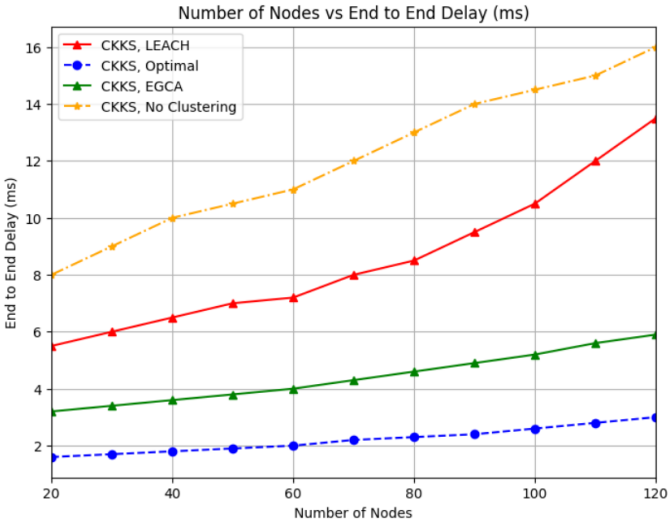

Fig. 3. End to end delay as a function of the number of nodes for different clustering configurations

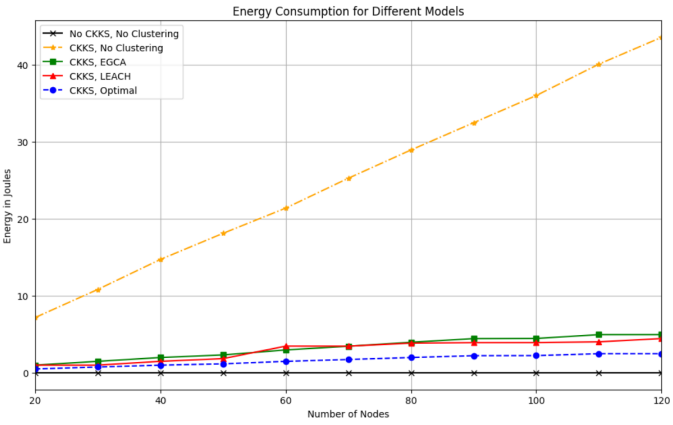

Fig. 4. Energy consumption comparison across different clustering configurations and encryption models

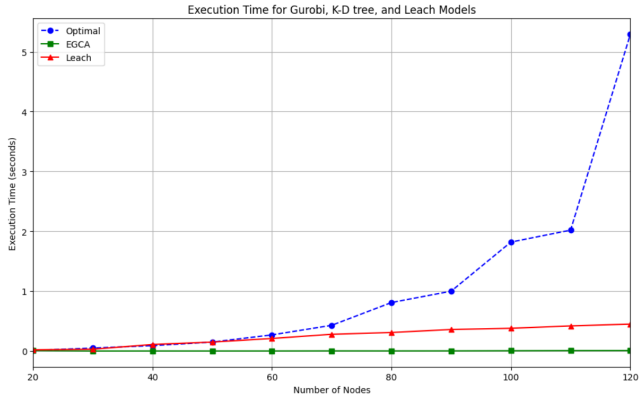

Fig. 5. Execution time comparison for Gurobi, EGCA, and LEACH as network size increases

increases. The Gurobi solver, while providing optimal solutions, demonstrates a sharp rise in execution time, exceeding 5 seconds for larger networks, making it inefficient for real-time re-optimization in large-scale scenarios. In contrast, EGCA maintains consistently low execution times, showcasing its scalability and efficiency in generating near-optimal clustering solutions. Although faster than Gurobi, LEACH exhibits a gradual increase in execution time as the number of nodes grows, performing slightly slower than EGCA. These results underscore EGCA's suitability as a more scalable and practical approach for real-time applications than Gurobi and LEACH.

VI. Conclusion

In this paper, we addressed the challenge of optimizing energy consumption in sensor networks by developing an approach that organizes sensors into energy-efficient groups managed by designated nodes. This structure minimizes direct transmissions, reducing energy consumption and making CKKS homomorphic encryption more feasible for battery-powered sensors. Our simulations, comparing this method with LEACH and the Gurobi-based optimal solution, demonstrated near-optimal performance, with significant improvements in energy efficiency, Packet Delivery Ratio, and network reliability. These results highlight the effectiveness of our approach in smart agriculture, where energy conservation and robust security are critical.

VII. Acknowledgement

The research is supported by Ali Jalooli's grant no. DHS-23STSLA00019. However, any opinion, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DHS.

References

[1] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie, and T. Shongwe, "Internet of things and wireless sensor networks for smart agriculture applications: A survey," *IEEE Access*, vol. 11, pp. 145 813–145 827, Dec. 2023.

[2] A. Yazdinejad, B. Zolfaghari, A. Azmoodeh, A. Dehghantanha, H. Karimipour, E. Fraser *et al.*, "A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures," *Applied Sciences*, vol. 11, no. 16, p. 7518, 2021.

[3] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017*, ser. Lecture Notes in Computer Science. Springer, 2017, vol. 10624, pp. 409–437.

[4] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023. [Online]. Available: https://www.gurobi.com

[5] M. Zhou, Y. Zheng, Y. Guan, L. Peng, and R. Lu, "Efficient and privacy-preserving range-max query in fog-based agricultural iot," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2156–2170, 2021.

[6] G. Singhal and A. Singh, "Data storage, collection, and transmission in smart agriculture using bloom filter," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2023, pp. 1–7.

[7] N. H. Nguyen, T. A. Nguyen, T. Nguyen, V. T. Hoang, D. D. Le, and K. S. Wong, "Towards efficient communication federated recommendation system via low-rank training," *arXiv preprint arXiv:2401.03748*, 2024.

[8] A. Vangala, A. K. Das, V. Chamola, V. Korotaev, and J. J. Rodrigues, "Security in iot-enabled smart agriculture: Architecture, security solutions and challenges," *Cluster Computing*, vol. 26, no. 2, pp. 879–902, 2023.

[9] K. Taji and F. Ghanimi, "A privacy-preserving robust and efficient homomorphic signcryption system tailored for smart agriculture," *Data and Metadata*, vol. 2, p. 112, dec 2023. [Online]. Available: https://doi.org/10.56294/dm2023112

[10] O. A. Khashan, N. M. Khafajah, N. Mohamed, and B. Salaimah, "Energy-efficient dynamic adaptive encryption for low-resource internet of things," in *EAI 3rd International Conference on Smart Technologies and Innovation Management*. Cham: Springer Nature Switzerland, 2025, pp. 191–202.

[11] O. A. Khashan, N. M. Khafajah, W. Alomoush, and M. Alshinwan, "Innovative energy-efficient proxy re-encryption for secure data exchange in wireless sensor networks," *IEEE Access*, vol. 12, pp. 23 290–23 304, 2024.

[12] E. R. P. and M. D.S., "Lightweight key distribution for secured and energy efficient communication in wireless sensor network: An optimization assisted model," *High-Confidence Computing*, vol. 3, no. 2, p. 100126, 2023.

[13] M. Tong and M. Tang, "Leach-b: an improved leach protocol for wireless sensor network," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*. IEEE, 2010, pp. 1–4.