Achieving $O(\log^3 n)$ Communication-Efficient Privacy-Preserving Range Query in Fog-Based IoT

Hassan Mahdikhani, Rongxing Lu, *Senior Member, IEEE*, Yandong Zheng, Jun Shao, and Ali Ghorbani, *Senior Member, IEEE*

***Abstract*—The advance of Internet of Things (IoT) techniques has promoted an increasing number of organizations to explore more mission-critical solutions. However, the response latency, bandwidth usage, and reliability are still challenging issues in traditional IoT. To tackle these challenges, fog-based IoT has become popular and range query is one of the most frequently used operations in fog-based IoT, where, given a range query, a fog node will return the aggregated data from IoT devices to the query user. Because the fog nodes are not fully trusted, there is a desire to design a privacy-preserving range query scheme in fog-based IoT. However, most of existing privacy-preserving range query schemes are not efficient in terms of communication overhead, especially for a large-size range. Therefore, it is still a challenging issue to design a communication-efficient range query in fog-based IoT. Aiming at this challenge, in this paper, we propose a new privacy-preserving range query scheme in fog-based IoT. Specifically, we first devise an efficient homomorphic encryption scheme for maintaining the data privacy and security in a range query. Then, we present a novel range decomposition technique to compile the range query, which can transform a given range query $[L, U]$, where $0 \leq L \leq U \leq n-1$, into a semi-triangular structure, and enable our proposed scheme to achieve $O(\log^3 n)$ communication efficiency. Detailed security analysis shows that our proposed scheme is really privacy-preserving, and the extensive performance evaluation demonstrates that our proposed scheme is efficient in terms of low communication overhead and computational cost.**

***Index Terms*—Fog-based IoT, range query, privacy-preserving, communication efficiency.**

I. Introduction

With ongoing advances in research and development of the Internet of Things (IoT), the connected IoT devices are deployed in an ever-increasing number of projects in different fields [1]–[4], varying from cryptocurrencies (IOTA Tangle [5]) to even food safety industries [6] and smart grid [7]–[9] and they generate huge volumes of data. As reported in [10], there will be more than 10 billion IoT devices interconnected via the internet by 2020, and all these devices will generate around 4.4 trillion GB of data. As a result, IoT emerging technologies will be expected to investigate in the decades ahead, as they have been broadly studied over the past few years. In conventional IoT applications, IoT devices are employed to gather distributed data from the operational environment and transmit them to central decision-making points in the cloud. Obviously, in outsourcing time-sensitive big data applications, certainly there exist some potential concerns, such as the response latency, bandwidth usage, and reliability that should be carefully considered [11]. To this end, fog-based IoT has been emerged to address the aforementioned factors where the big data is being analyzed. The response is aggregated closer to where the IoT devices are deployed [12]–[14]. At the same time, the fog extends the conventional cloud to the edge of the network providing real-time or at least fast enough computation, and also efficient storage and networking services, between the IoT devices and cloud layer/users [15]. However, since the fog nodes are deployed at the network edge, they may not be fully trusted. As a result, privacy-preserving techniques are essential in fog-based IoT scenarios.

In this paper, we will devise an efficient privacy-preserving range query scheme in fog-based IoT. In our system, there are N IoT devices $\{I_i | 1 \leq i \leq N\}$, and each I_i is preparing a reported data $x_i \in [0, n-1]$. For the query user, he/she intends to submit his/her range query request to a fog node at the network edge. The range query is in the form of $[L, U]$, which means that the user queries "*the number of IoT devices whose data x_i is within the range [L,U], where* $0 \leq L \leq U \leq n-1$". A potential application scenario for this kind of range query is a smart grid, where a smart grid operator uses the query to track the finer-grained electricity consumption of a neighborhood for better scheduling and optimization purposes [16]. Aiming to achieve desired privacy goals, the query user expects to keep the range values, i.e., L and U, confidential, and each I_i tries to keep its individual data x_i secure. A naive approach towards such privacy-preserving range query is to use homomorphic encryption (HE) schemes such as BGN [17] and Paillier [18]. Suppose that $n = 10$, and there are $N = 20$ IoT devices $\{I_i | 1 \leq i \leq 20\}$ and each I_i is preparing its data x_i, where $0 \leq x_i \leq n-1 = 9$. At the same time, the user intends to run the range query with corresponding parameters $L = 2$, and $U = 6$. With the naive solution, the user first represents the query range $[L, U]$ as an array $A = \{0, 0, 1, 1, 1, 1, 1, 0, 0, 0\}$ of size n. $A[i]$ is set to one if $L \leq i \leq U$ and otherwise it is set to zero. Then, he/she encrypts A with the HE encryption algorithm $Enc()$ as $Enc(A) = \{Enc(A[0]), Enc(A[1]), \cdots, Enc(A[9])\}$. After that, he/she sends $Enc(A)$ to the fog node which forwards $Enc(A)$ to each IoT device. Upon receiving $Enc(A)$ from the fog node, each IoT device I_i with its ready-to-report value x_i calculates $c_i \leftarrow \text{selfBlind}(Enc(A[x_i]))$ and sends its c_i to the fog node. Note that, the *selfBlind* is a nice property

H. Mahdikhani, R. Lu, Y. Zheng, and A. Ghorbani are with the Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada E3B 5A3. e-mail: (hmahdikh@unb.ca, rlu1@unb.ca, yzheng8@unb.ca, ghorbani@unb.ca).

J. Shao is with School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China 310018 e-mail: (chn.junshao@gmail.com).

Manuscript received December xx, 2019.

of HE for better privacy preservation, where a ciphertext can be transformed into another one without changing the corresponding plaintext [19], [20]. Finally, the fog node aggregates the ciphertexts received from the IoT devices into a single ciphertext, i.e., $C = \sum_{i=1}^{N} c_i$ and then sends C to the query user. Although the above naive solution can satisfy the privacy concerns in the range query, its communication cost is prohibitively large, i.e., $O(n)$. In order to reduce the communication cost, Lu [16] proposed a privacy-preserving range query scheme with $O(\sqrt{n})$ communication efficiency, which is acceptable for small values of n, but cannot support large-size n. Therefore, designing a communication efficient privacy preserving range query scheme in fog-based IoT is still challenging, especially for a large n value.

Aiming at the above challenges, in this paper, we propose a new efficient privacy-preserving range query scheme with $O(\log^3 n)$ communication efficiency for fog-based IoT. More precisely, our main contributions in this paper are summarized as follows:

- First, we design a novel decomposition technique, which can transform a given range query $[L, U]$ into several space efficient semi-triangular structures, and further turns them into sparse matrices for keeping the privacy of the query.
- Second, we propose a symmetric homomorphic encryption scheme, which can support $b = \log_2 n$ times multiplicative homomorphic operations. The proposed homomorphic encryption (SHE) scheme not only offers different homomorphic operations, but also provides an efficient and secure way to prepare, disseminate, and calculate ciphertexts.
- Third, based on the decomposition technique and the proposed symmetric homomorphic scheme, we present an efficient privacy-preserving range query scheme with $O(\log^3 n)$ communication efficiency for fog-base IoT.
- Finally, we analyze the security of our proposed scheme and evaluate its performance. The results show that our proposed scheme can indeed satisfy the desired privacy concerns, and outperforms the most recently developed approach in [16] in terms of computational costs and communication overhead.

The remainder of the paper is structured as follows. The system and security models along with design goal are introduced in Section II. Section III contains the description of our symmetric homomorphic encryption scheme and its properties. Then, Section IV presents our proposed scheme, followed by security analysis in Section V. Furthermore, the detailed performance evaluation is organized in Section VI. Some related works are discussed in Section VII and, finally, conclusions are drawn in Section VIII.

II. Models and Design Goal

In this section, we formalize our system model, security model, and identify the design goal for our communication-efficient privacy-preserving range query. For the clear description, some used symbols are first listed in Table I.

TABLE I
THE LIST OF SYMBOLS USED IN PROPOSED SCHEME

Symbol	Description
N	The number of the IoT devices.
I_i	The i-th IoT device in set $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$, $1 \le i \le N$.
n	The maximum possible value that can be sensed by IoT devices.
x_i	Sensed and prepared data in I_i range from 1 to n; $1 \le x_i \le n$.
L, U	Lower and upper bound in range query, $1 \le L = 2^a \le U = 2^{a'} \le n$.
$\mathbf{I}'$	The subset of I where I_i's data $x_i \in [L, U]$.
PP	Public parameter in SHE scheme; $PP = (k_0, k_1, k_2, \mathcal{N})$.
SK	Secure key in SHE scheme; $SK = (p, q, \mathcal{L})$.
$Enc(m)$	Encryption method in cryptosystem, $m \in \mathcal{M} \in \{0,1\}^{k_1}$.
$Dec(c)$	Decryption method in cryptosystem, $\lvert c\rvert = \lvert\mathcal{N}\rvert = 2k_0$.
B	Range's binary representation, $\lvert B\rvert = U - L + 1$.
S	Hamming weight classification of B, $\lvert S\rvert = \log_2 n + 1$=b+1.
C	Cumulative sum values for each entry in S, $\lvert C\rvert = \lvert S\rvert$.
R_i	Semi-triangular matrix generated from each C_i.
T_i	The corresponding sparse matrix for each R_i.

A. System Model

Fig. 1 illustrates our system model from a high-level perspective. It consists of three major entities, namely IoT devices $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$, a fog node in the network edge and a query user.

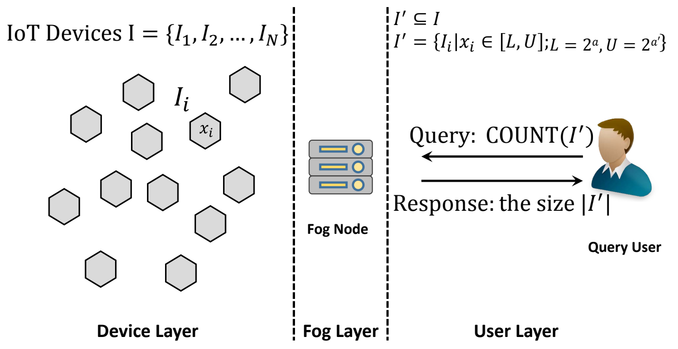

IoT Devices I = $\{I_1, I_2, \dots, I_N\}$
$I' \subseteq I$
$I' = \{I_i | x_i \in [L, U]; L = 2^a, U = 2^{a'}\}$
I_i
x_i
Query: COUNT(I')
Response: the size $|I'|$
Fog Node
Query User
Device Layer
Fog Layer
User Layer

Fig. 1. System model under consideration

- IoT devices $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$: A myriad of IoT devices $\mathbf{I}$ of size N are remotely spread across the device layer and get involved in a specific IoT solution. Each IoT device I_i gathers raw data x_i from the physical sensing environment and will typically perform some processing tasks on them. The prepared data x_i afterwards will be reported to the fog node in the upper layer for a later processing. It is worth noting that in our system model the reported data will be converted into an integer in the range of $[0, n-1]$ to clarify the processing steps and declaring the scheme. To this end, we can easily transform floating-point values into integers by applying simple scaling, truncating or rounding functions. However, using a common conversion method(s) or combination of them depends on what kind of conversion is desired and what works best, e.g., $x_i = 198.73549$ can be transformed into an integer value by $truncate(x_i = 198.73549 * 10) = 1987$, if the upper bound value for n is $2^{11} = 2048$; and $x_i = 1987354$ for a scaling factor of 10000 where $n = 2^{21} = 2097152$. Therefore, accepting large scaling factors and consequently large n values, which our model supports, will lead to more accurate results. More precisely, our proposed scheme in this work would be more effective for large n values, i.e., $n \geq 2^{17}$, and much more efficient than a recent related study in [16].

• Fog node: The fog layer, as an intermediate layer, comprises various communication and computing elements, namely fog nodes, which handle a near real-time stream of requests and responses. In our system model, the fog node is a computational entity more powerful than IoT devices but not as powerful as cloud entities. After receiving prepared data x_i from lower layer IoT device I_i, the fog node applies filtering rules and performs aggregation operations. Afterwards, the aggregated data will be stored in local storage or rerouted to the upper layer for further analysis. A query user in the user layer, according to his/her query preference, will submit a range query to fog layer and shortly receive the reply from the fog node.

• Query user: As depicted in Fig 1, a query user can submit a secure range query to fetch the number of IoT devices whose prepared data x are within the range $[L, U]$, where L and U are respectively the lower and upper bounds of the range, and are both in the form of "two to the power of a positive integer", i.e., $L = 2^a \leq U = 2^{a'}$. Suppose that $\mathbf{I}'$ is a subset of $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$ where the prepared data x_i of each IoT device $I_i \in \mathbf{I}'$ is within the range $[L, U]$, i.e.,

$$\mathbf{I}' = \{I_i | I_i \in \mathbf{I} \wedge x_i \in [L, U]; L = 2^a, U = 2^{a'}\} \tag{1}$$

The query user will then submit the following range count query to find the number of IoT devices whose x_i is within the given range, i.e., Query: COUNT($\mathbf{I}'$) $\longrightarrow$ Response: the size $|\mathbf{I}'|$.

B. Security Model

In our security model, all entities, deployed at both device and fog layers, are assumed to be honest-but-curious participants, i.e., they are obliged to follow the protocols faithfully, but may be curious about the query information during data preparation and throughout the query processing steps. For example: i) the fog node may try to identify the IoT devices I_i with exact data x_i, ii) each IoT device may be curious about other IoT devices' data, iii) both fog node and IoT devices may try to detect the lower (L) and/or upper (U) bound values, and finally, iv) the user may be curious about each IoT device I_i's prepared data or, at least, prepared data in each IoT device in $\mathbf{I}'$. Note that the honest-but-curious assumption would be guaranteed in practice, since the service providers should protect their own reputation and financial interests. In addition, there would be no collusion between entities, such as sharing their results. Lastly, to mitigate external active attacks, some mature message authentication code techniques or digital signature schemes could be applied. However, considering such potential attacks are beyond the scope of this paper and we focus on the communication efficient privacy-preserving range query.

C. Design Goal

Based on the aforementioned requirements in both system model and security model, our goal is to have an efficient privacy-preserving range query scheme with $O(\log^3 n)$ communication efficiency in a fog-based IoT environment. Specifically, the following two objectives should be satisfied.

• *The proposed scheme should be privacy-preserving.* In the proposed scheme, the user's query range $[L, U]$ should be privacy-preserving, i.e., no one, except the user, can determine $[L, U]$. In addition, the elements of subset $\mathbf{I}'$ should also be privacy-preserving, i.e., no one can determine whether a specific IoT device belongs to $\mathbf{I}'$ or not, and only the query user can know COUNT($\mathbf{I}'$) after the range query.

• *The proposed scheme should be communication efficient.* With the intention of addressing the above privacy objectives in range queries, additional communication overhead will be incurred in the range query, as we have discussed in Section I. Therefore, in the proposed scheme, we attempt to improve the communication efficiency by achieving $O(\log^3 n)$ communication cost, which is better than the current most communication efficient scheme with $O(\sqrt{n})$ communication cost in [16].

III. Preliminaries

In order to build our communication-efficient privacy-preserving range query scheme, we need to adopt homomorphic encryption techniques, which can support both homomorphic addition and multiplication. However, most existing homomorphic encryption schemes use public key settings, which are not computationally efficient. For this reason, in this section, we first introduce an efficient symmetric homomorphic encryption (SHE) scheme, which will serve as the fundamental building block for our privacy-preserving range query.

A. Description of SHE Scheme

As a symmetric homomorphic encryption scheme, SHE mainly consists of three algorithms, namely *key generation*, *encryption*, and *decryption*. In the following, we describe them in detail.

- *Key Generation:* Given the security parameters (k_0, k_1, k_2) satisfying $k_1 \ll k_2 < \frac{k_0}{2}$, generate the secret key $SK = (p, q, \mathcal{L})$, where p, q are two large prime numbers with $|p| = |q| = k_0$ and $\mathcal{L}$ is a random number with the bit length $|\mathcal{L}| = k_2$. Then, compute $\mathcal{N} = pq$ and set the public parameter $PP = (k_0, k_1, k_2, \mathcal{N})$. At the same time, set the message space $\mathcal{M}$ as $\{0, 1\}^{k_1}$.
- *Encryption:* A message $m \in \mathcal{M}$ can be encrypted with the secret key $SK = (p, q, \mathcal{L})$ as

  $$c = Enc(m) = (r\mathcal{L} + m)(1 + r'p) \bmod \mathcal{N} \tag{2}$$

  where $r \in \{0, 1\}^{k_2}$ and $r' \in \{0, 1\}^{k_0}$ are two random numbers.
- *Decryption:* A ciphertext $c = Enc(m)$ can be decrypted with the secret key $SK = (p, q, \mathcal{L})$ as

  $$Dec(c): \; m = (c \bmod p) \bmod \mathcal{L} \tag{3}$$

The correctness of the decryption is as follows.

$$\begin{aligned} Dec(c) &= (c \bmod p) \bmod \mathcal{L} \\ &= (((r\mathcal{L}+m)(1+r'p) \bmod \mathcal{N}) \bmod p) \bmod \mathcal{L} \\ &= (r\mathcal{L}+m) \bmod \mathcal{L} \quad (\because 2k_2 < k_0) \\ &= m \quad (\because k_1 \ll k_2) \end{aligned}$$

Security of SHE. Obviously, our SHE scheme is simple and efficient. Next, we will prove that it is also secure. Let $M = r\mathcal{L} + m$, we know $M < p$, and then the ciphertext $c = Enc(m) = (r\mathcal{L}+m)(1+r'p) \bmod \mathcal{N}$ becomes

$$c = M \cdot (1 + r'p) \bmod \mathcal{N} \tag{4}$$

Theorem 1. *Given the ciphertext $c = M \cdot (1+r'p) \bmod \mathcal{N}$, obtaining the message M is equivalent to factoring the larger integer $\mathcal{N} = pq$.*

Proof: First, if $\mathcal{N} = pq$ can be factored into p and q, from the decryption algorithm, we can use p to obtain $M = c \bmod p$. Second, if we can obtain M from $c = M\cdot(1+r'p) \bmod \mathcal{N}$, we know $r'p = (c\cdot M^{-1}-1) \bmod \mathcal{N}$. Then, from the fact that $\gcd(r'p, \mathcal{N}) = p$, we can factor the larger integer $\mathcal{N} = pq$. Therefore, obtaining the message M from c is equivalent to factoring the larger integer $\mathcal{N} = pq$. Only if the larger integer factoring problem is hard with proper parameter settings, we can make sure that the message M is secure. ■

Theorem 2. *Our proposed SHE scheme is secure under the known-plaintext attack (KPA).*

Proof: Given a pair (c, m), a ciphertext $c = Enc(m) = (r\mathcal{L}+m)(1+r'p) \bmod \mathcal{N}$ and its corresponding plaintext m, we will show that an adversary still cannot obtain the secret key $SK = (p, q, \mathcal{L})$. First, in order to obtain the secret key (p, q), according to Theorem 1, it is equivalent to obtain the message $M = r\mathcal{L} + m$ from (c, m). Since the bit-length of the whole $r\mathcal{L}$ is $|r\mathcal{L}| = 2k_2$, we know the adversary can guess the correct $r\mathcal{L}$ and $M = r\mathcal{L} + m$ only with probability $\frac{1}{2^{2k_2}}$. Therefore, only if the parameter k_2 is well set, the secret key (p, q) cannot be obtained from (c, m). Second, the adversary can directly guess the correct secret key $\mathcal{L}$ with probability $\frac{1}{2^{k_2}}$. But he still needs to guess the correct r with probability $\frac{1}{2^{k_2}}$ so as to guess the correct $r\mathcal{L}$ and $M = r\mathcal{L}+m$. Therefore, guessing $\mathcal{L}$ with probability $\frac{1}{2^{k_2}}$ is not helpful for guessing M and the secret key (p, q) with probability $\frac{1}{2^{2k_2}}$. As a result, our proposed SHE scheme is secure under the known-plaintext attack (KPA). ■

B. Homomorphic Properties of SHE

Given the public parameter PP, our proposed SHE scheme enjoys the following homomorphic properties:

- *Homomorphic Addition-I:* Given two ciphertexts $c_1 = Enc(m_1) = (r_1\mathcal{L} + m_1)(1 + r_1'p) \bmod \mathcal{N}$, $c_2 = Enc(m_2) = (r_2\mathcal{L} + m_2)(1 + r_2'p) \bmod \mathcal{N}$, we have $c_1 + c_2 \rightarrow Enc(m_1 + m_2)$. This is because

$$\begin{aligned} & c_1 + c_2 \\ &= (r_1\mathcal{L}+m_1)(1+r_1'p) + (r_2\mathcal{L}+m_2)(1+r_2'p) \bmod \mathcal{N} \\ &= (r_1+r_2)\mathcal{L} + m_1 + m_2 + \alpha \cdot p \bmod \mathcal{N} \\ \Rightarrow & ((c_1 + c_2) \bmod p) \bmod \mathcal{L} \\ &= ((r_1+r_2)\mathcal{L} + m_1 + m_2) \bmod \mathcal{L} = Enc(m_1 + m_2) \end{aligned}$$

  where $\alpha = (r_1r_1' + r_2r_2')\mathcal{L} + (m_1r_1' + m_2r_2')$.
- *Homomorphic Multiplication-I:* $c_1 \cdot c_2 \rightarrow Enc(m_1 \cdot m_2)$. This is because

$$\begin{aligned} & c_1 \cdot c_2 \\ &= (r_1\mathcal{L}+m_1)(1+r_1'p) \cdot (r_2\mathcal{L}+m_2)(1+r_2'p) \bmod \mathcal{N} \\ &= \beta'\mathcal{L} + m_1 \cdot m_2 + \alpha' \cdot p \bmod \mathcal{N} \\ \Rightarrow & ((c_1 \cdot c_2) \bmod p) \bmod \mathcal{L} \\ &= (\beta'\mathcal{L} + m_1 \cdot m_2) \bmod \mathcal{L} = Enc(m_1 \cdot m_2) \end{aligned}$$

  where α' and β' are derived from $c_1 \cdot c_2$.
- *Homomorphic Addition-II:* Given a ciphertext $c_1 = Enc(m_1) = (r_1\mathcal{L}+m_1)(1+r_1'p) \bmod \mathcal{N}$, and a plaintext m_2, we have $c_1 + m_2 \rightarrow Enc(m_1 + m_2)$. This is because

$$\begin{aligned} & c_1 + m_2 \\ &= (r_1\mathcal{L}+m_1)(1+r_1'p) + m \bmod \mathcal{N} \\ &= r_1\mathcal{L} + m_1 + m_2 + (r_1\mathcal{L}+m_1) \cdot p \bmod \mathcal{N} \\ \Rightarrow & ((c_1 + m_2) \bmod p) \bmod \mathcal{L} \\ &= (r_1\mathcal{L} + m_1 + m_2) \bmod \mathcal{L} = Enc(m_1 + m_2) \end{aligned}$$

- *Homomorphic Multiplication-II:* $c_1 \cdot m_2 \rightarrow Enc(m_1 \cdot m_2)$. This is because

$$\begin{aligned} & c_1 \cdot m_2 \\ &= (r_1\mathcal{L}+m_1)(1+r_1'p) \cdot m_2 \bmod \mathcal{N} \\ &= r_1m_2\mathcal{L} + m_1 \cdot m_2 + (r_1\mathcal{L}+m_1)m_2 \cdot p \bmod \mathcal{N} \\ \Rightarrow & ((c_1 \cdot c_2) \bmod p) \bmod \mathcal{L} \\ &= (r_1m_2\mathcal{L} + m_1 \cdot m_2) \bmod \mathcal{L} = Enc(m_1 \cdot m_2) \end{aligned}$$

Based on the above homomorphic properties of SHE, we will design our communication-efficient privacy-preserving range query scheme in next section.

IV. Our Proposed Scheme

In this section, we will describe our privacy-preserving range query scheme in fog-based IoT. Before delving into the details, we first present our decomposition technique, which is a building block in our proposed scheme.

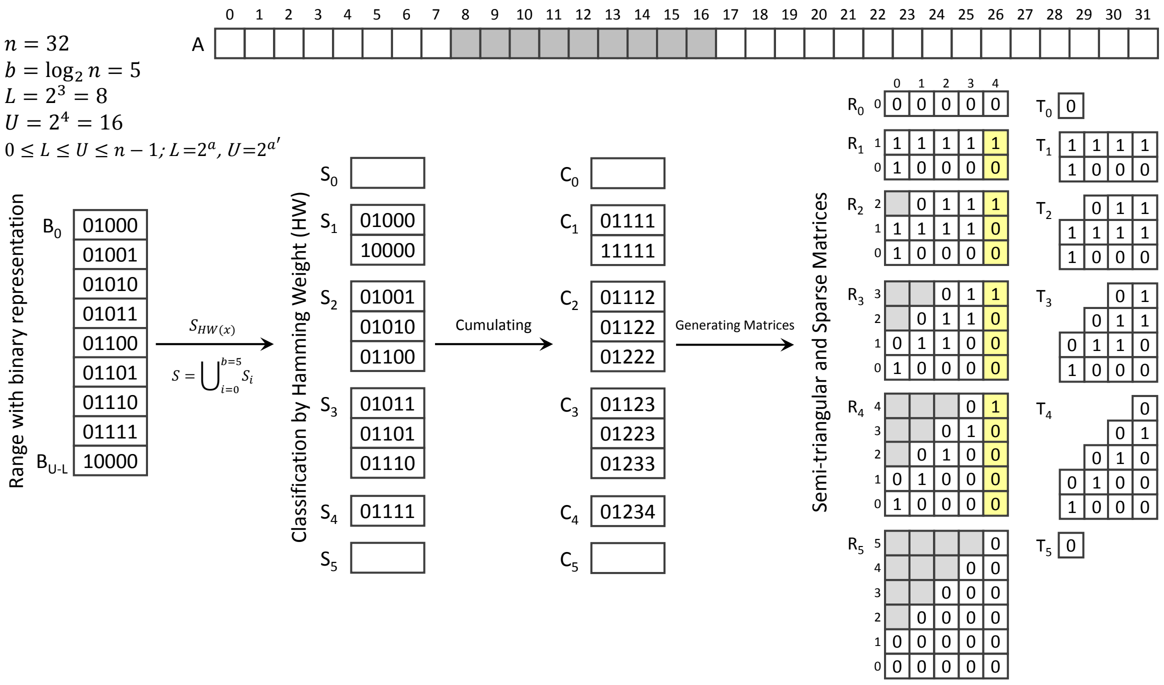
A. The Decomposition Technique

Our decomposition technique can transform a query range $[L = 2^a, U = 2^{a'}]$ into $(\log_2 n + 1)$ sparse matrices, where $a \leq a'$ and $0 \leq L \leq U \leq n-1$ and consists of five steps, 1) range query formulation, 2) binary representation of the range, 3) Hamming-weight classification, 4) cumulative sums structures construction, and finally 5) semi-triangular matrices generation. Note that, for a clear description, we also take the values $n = 32$, $[L = 2^3, U = 2^4]$ and instantiate a simple example in Fig. 2, in which 6 sparse matrices $\{T_0, \cdots, T_5\}$ are generated, since $\log_2 n + 1 = \log_2 32 + 1 = 6$.

1) Range Query Formulation: Given the lower bound L and upper bound U, the general syntax of a range query statement can be outlined using an array $A[0..n-1]$ as follows,

$$A[i] = \begin{cases} 1, & L \leq i \leq U; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $0 \leq L \leq U \leq n-1$ and $0 \leq i \leq n-1$.



$n = 32$
$b = \log_2 n = 5$
$L = 2^3 = 8$
$U = 2^4 = 16$
$0 \le L \le U \le n-1; L=2^a, U=2^{a'}$
A
Range with binary representation
B_0
B_{U-L}
$S_{HW(x)}$
$S = \bigcup_{i=0}^{b=5} S_i$
Classification by Hamming Weight (HW)
Cumulating
Generating Matrices
Semi-triangular and Sparse Matrices

Fig. 2. Converting a range query into semi-triangular and sparse matrices. For example, the corresponding range query $[L, U] = [2^3, 2^4]$ of array A where $L \le U \le 31$ is transformed into $(\log_2 n) + 1 = 6$ sparse matrices $T_0, T_1, \cdots, T_{\log_2 n}$.

2) Binary Representation of the Range: In this step, the given range query $[L, U]$ is converted into a vector of bit-strings. As shown in Algorithm 1, for each $i \in [L, U]$, generate the equivalent binary sequence of i by simply applying two bitwise operations *RightShift* and *AND*. Clearly, the expected output B is a vector of binary strings where $|B| = U - L + 1$ and each entry B_ℓ, where $0 \le \ell \le U - L$, of B is a string of zeros and ones of length b, i.e., $B_\ell \in \{0, 1\}^b$ and $b = \log_2 n$, as shown in Fig. 2.

Algorithm 1 Range's Binary-Representation

```
Input: $b = \log_2 n$, $L$, and $U$; $0 \le L = 2^a \le U = 2^{a'} \le n - 1$
Output: Vector $<B>$; $|B| = U - L + 1$, and $|B_\ell \in B| = b$
1: $B \leftarrow \emptyset$; $\ell \leftarrow 0$                    $\triangleright$ $0 \le \ell \le U - L$
2: for each $i \in [L, U]$ do
3:     $c \leftarrow b - 1$
4:     $B_\ell \leftarrow$ “”
5:     while $c \ge 0$ do
6:         $B_\ell$.APPEND$\Big($RIGHTSHIFT$(i, c)$ & $1\Big)$
7:         $c \leftarrow c - 1$
8:     $\ell \leftarrow \ell + 1$
9: return $B$
```

3) Hamming Weight Classification: The Hamming weight of a given bit-string x is the Hamming distance (HD) of two equal-length bit-strings x and zero-codewords; i.e., $HD(x, \{0\}^{|x|})$. In other words, the Hamming weight of a given bit-string x is the number of ones in x and it can be computed by the HAMMINGWEIGHT() procedure in Algorithm 2. For example, the Hamming weight of B_0 in Fig. 2, can be denoted as $HD($“01000”, “00000”$) = 1$. Therefore, all the input bit-strings in vector B will be categoried into $\log_2 n + 1$ classes to form vector S as shown in Algorithm 2, as well as $S = (S_0, S_1, \cdots, S_5)$ in our simple example in Fig. 2, which will be used to generate the cumulative sum structure in the next step.

Algorithm 2 Hamming Weight Classification

```
Input: Vector $<B>$
Output: Vector <Vector $<S>$> and $|S| = \log_2 n + 1 = b + 1$
1: $S \leftarrow \emptyset$
2: for each $B_\ell \in B$ do
3:     $\omega \leftarrow$ HAMMINGWEIGHT$(B_\ell)$                $\triangleright$ $0 \le \omega \le b$
4:     $S_\omega$.ADD$(B_\ell)$
5: return $S$

6: procedure HAMMINGWEIGHT$(x)$
7:     $t \leftarrow 0$
8:     while $x \ne 0$ do
9:         $x = x$ & $(x - 1)$
10:        $t \leftarrow t + 1$
11:    return $t$
```

4) Cumulative Sum Structure Construction: As shown in Algorithm 3, in this step, our goal is to generate the cumulative sum structure for each entry in S, i.e., $\forall s_j \in S_i, 0 \le i \le b$. As seen in lines 6-11 of the algorithm, a cumulative sum t is a same-length sequence of partial sums of a given bit sequence x or clearly the value at each index of t equals the sum of current and all the preceding values. For example, as shown in Fig. 2, the cumulative sum $c_0 \in C_3 =$ “$0, 1, 1, 2, 3$” for $s_0 \in S_3 =$ “01011” can be interpreted as: $c_{00} = s_{00} = 0, c_{01} = s_{01} + c_{00} = 1 + 0 = 1, c_{02} = s_{02} + c_{01} = 0 + 1 = 1, c_{03} = s_{03} + c_{02} = 1 + 1 = 2$, and finally $c_{04} = s_{04} + c_{03} = 1 + 2 = 3$. It should be noted that, unlike the B's and S's elements, each $c_j \in C_i$ is itself a vector of integer values instead of bits to store the partial sums.

5) Semi-triangular Matrices Generation: Finally, Algorithm 4 converts the cumulative sum structure C into semi-

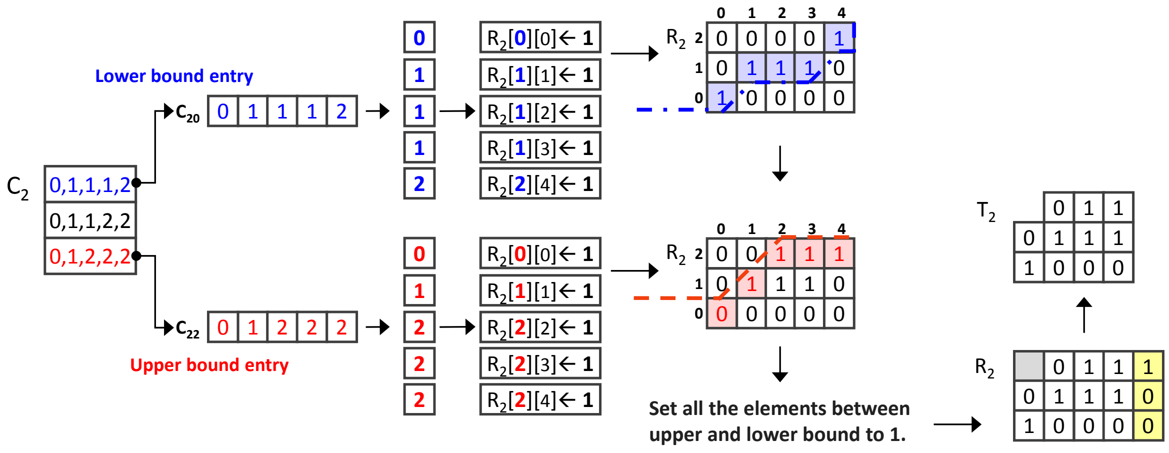


Fig. 3. Generating matrix R_2 and sparse structure T_2 from cumulative sum C_2

Algorithm 3 Cumulative Sum Structure

```
Input: $S = \{S_i\} = \{\{s_j | \text{HAMMINGWEIGHT}(s_j) = i\}\}$
Output: Vector < Vector <Vector <C>>> and $|C| = |S|$
 1: $C \leftarrow \emptyset$
 2: for each $S_i \in S$ do                              ▷ $0 \le i \le b$
 3:     for each $s_j \in S_i$ do
 4:         $C_{ij}$.ADD(CUMMULATIVESUM($s_j$))           ▷ $j = 0, 1, 2, \cdots$
 5: return $C$

 6: procedure CUMMULATIVESUM($x$)
 7:     Vector <t> ← ∅
 8:     $t$.ADD($x_0$)
 9:     for $i = 1$ to LENGTH($x$) − 1 do
10:         $t$.ADD($x_i + t$.GET($i - 1$))
11:     return $t$
```

triangular matrices. For clarity, we take the cumulative sum C_2 in Fig. 2 as an example to show how to transform C_2 to a semi-triangular matrix R_2, as depicted in Fig. 3. Before analyzing the example, it should be noted that the result matrix R_i has $i+1$ rows and $b = \log_2 n$ columns and is initialized as a zero matrix at the beginning of the process. Meanwhile, the row index numbering starts from zero and is increased from bottom to up, and analogously the column index starts from zero and is numbered from left to right. Therefore, the result matrix R_2 has $3 = 2+1$ rows and $5 = b = \log_2 32$ columns. In our detailed example in Fig. 3, the cumulative sum C_2 consists of three b-dimensional entries, i.e., c_{20}, c_{21}, and c_{22}. In the following, we show how to generate R_2 from C_2 step by step.

Step 1: Choose the lower and upper bound entries in C_2, i.e., the first entry c_{20} and the last entry c_{22}, and generate matrices for them. In specific, with c_{20} = "0, 1, 1, 1, 2", we can generate the lower bound entries in R_2 by updating

$$R_2\Big[c_{20}[k]\Big]\Big[k\Big] = \begin{cases} 1, & 0 \le k \le 4 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Similarly, with c_{22}, we generate the upper bound entries in R_2 by updating

$$R_2\Big[c_{22}[k]\Big]\Big[k\Big] = \begin{cases} 1, & 0 \le k \le 4 \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Step 2: Continue to generate the matrix R_i by setting all entries between the lower bound and upper bound to ones. After this step, we have $b+1$ matrices, i.e., $\{R_0, \cdots, R_b\}$. Note that, our strategy to only find the lower and upper bounds and fill all entries between them as ones in each matrix R_i. That speeds the generation of R_i, and its correctness for supporting the range query can be found in Theorem 3.

Step 3: Simplify each R_i to be a matrix T_i for $0 \le i \le b$, which can be divided into two cases.

- Case 1: For R_0 and R_b, since R_0 and R_b will only be set to 1 when the range query contains the lower bound value 0 and the upper bound value $n - 1 = 2^b - 1$; and set to 0 otherwise. Therefore, we can simplify R_0 and R_b to be 1×1 matrices T_0 and T_b. For example, in Fig. 2, both T_0 and T_5 are set to 0 because the given range query $[L, U] = [8, 16]$ does not include the lowest value 0 and the highest value 31.
- Case 2: For R_i ($1 \le i \le b-1$), since the last columns of R_i can be generated by simply combining a single one and i zeros, as shown in Fig. 2 and Fig. 3 (highlighted in yellow). Therefore, this entire column and related entries can be discarded to increase communication efficiency. Furthermore, in each matrix R_i, we can drop $1+2+3+\cdots+(i-1)$ entries, which are highlighted in grey, to obtain the corresponding matrix T_i.

Algorithm 4 Semi-triangular Matrices

```
Input: $C = \{C_i\} = \{\{c_j\}\} = \{\{\{Element_k\}\}\}$
Output: Vector <R[][]>
1: $R \leftarrow \emptyset$
2: for each $C_i \in C$ do                         ▷ $0 \le i \le b$
3:     $j \leftarrow \{0, C_i.\text{SIZE}()\}$              ▷ Lower and upper bound entries
4:     for each $Element_k \in c_j$ do              ▷ $0 \le k \le b-1$
5:         $t \leftarrow Element_k$
6:         $R_i[t][k] \leftarrow 1$
7:     for each LOWERBOUND < $R_i$[ ][ ] < UPPERBOUND do
8:         $R_i[t][k] \leftarrow 1$        ▷ $c_0 \le t \le c_{C_i.\text{SIZE}()} \wedge 0 \le k \le b-1$
9: return $R$
```

Theorem 3. *Our strategy employed for efficiently generating the matrices R_i, $i = 0, \cdots, b+1$, can support the range* *queries in the form of* $[L = 2^a, U = 2^{a'}]$*, where* $0 \le a \le a' \le b = \log_2 n$.

Proof: In order to prove the correctness of our strategy to support the range query $[2^a, 2^{a'}]$, we first prove that our strategy can support the range queries $[0, 2^a]$ and $[0, 2^{a'}]$.

For the range query $[0, 2^a]$, the parameter settings are $b = \log_2 n$; $[L, U] = [0, 2^a]$; the Hamming weight ω is within $0 \le \omega \le a$; and $0 \le a < b$. First, after the Hamming weight classification, the lower bound L_{S_ω} and upper bound U_{S_ω} for each class S_ω, where $0 \le \omega \le a$, are expressed as follows:

$$\omega = 0 \ \Rightarrow \text{lower } L_{S_0} = \text{upper } U_{S_0} = 0, \quad i.e., \quad \underbrace{0 \cdots 0}_{b}$$

$$\omega = \boldsymbol{\omega} \ \Rightarrow \begin{cases} \text{lower } L_{S_\omega} : \underbrace{0 \cdots 0}_{b-\boldsymbol{\omega}}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}} \\ \text{upper } U_{S_\omega} : \underbrace{0 \cdots 0}_{b-a}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}}\underbrace{0 \cdots 0}_{a-\boldsymbol{\omega}} \end{cases} \quad 1 \le \boldsymbol{\omega} \le a$$

Then, the corresponding cumulative sums are defined as

$$\omega = 0 \ \Rightarrow \text{lower } L_{C_0} = \text{upper } U_{C_0} = 0, \quad i.e., \quad \underbrace{0 \cdots 0}_{b}$$

$$\omega = \boldsymbol{\omega} \ \Rightarrow \begin{cases} \text{lower } L_{C_\omega} : \underbrace{0 \cdots 0}_{b-\boldsymbol{\omega}}\underbrace{123 \cdots \boldsymbol{\omega}}_{\boldsymbol{\omega}} \\ \text{upper } U_{C_\omega} : \underbrace{0 \cdots 0}_{b-a}\underbrace{123 \cdots \boldsymbol{\omega}}_{\boldsymbol{\omega}}\underbrace{\boldsymbol{\omega} \cdots \boldsymbol{\omega}}_{a-\boldsymbol{\omega}} \end{cases} \quad 1 \le \boldsymbol{\omega} \le a$$

For any value $x \in [0, n-1]$, if its Hamming weight $HW(x) = \omega$ is $\omega \le a$, then x must belong to the class S_ω. More accurately, it is easy to see that the value of x lies in the range $[L_{S_\omega}, U_{S_\omega}]$. As shown in Fig. 4, the cumulative sums $L_{C_\omega} = \underbrace{0 \cdots 0}_{b-\boldsymbol{\omega}}\underbrace{123 \cdots \boldsymbol{\omega}}_{\boldsymbol{\omega}}, U_{C_\omega} = \underbrace{0 \cdots 0}_{b-a}\underbrace{123 \cdots \boldsymbol{\omega}}_{\boldsymbol{\omega}}\underbrace{\boldsymbol{\omega} \cdots \boldsymbol{\omega}}_{a-\boldsymbol{\omega}}$ of S_ω's lower and upper bounds $L_{S_\omega}, U_{S_\omega}$ form a shaded parallelogram area. Because all elements in S_ω have the same weight ω, their cumulative sums will finally reach the same point $R_\omega[\omega][b] = 1$ in the matrix R_ω. Therefore, if x really belongs to $[L_{S_\omega}, U_{S_\omega}]$, its cumulative sum will be within the shaded parallelogram area in Fig. 4. As a result, our strategy of only finding the lower and upper bounds and filling all entries between them as ones in each matrix R_ω can support the range query with $[L, U] = [0, 2^a]$. Obviously, our strategy can also support the range query $[0, 2^{a'}]$. Next, we prove that our strategy can support the range query $[L = 2^a, U = 2^{a'}]$.

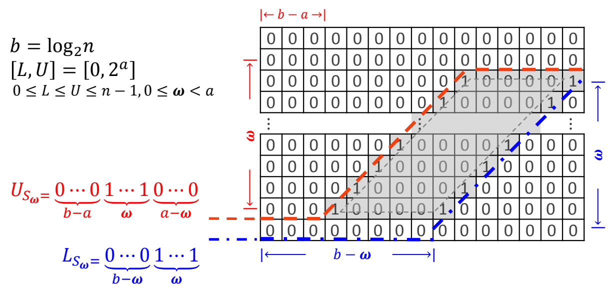

$b = \log_2 n$
$[L, U] = [0, 2^a]$
$0 \le L \le U \le n-1, 0 \le \omega < a$
$|\leftarrow b-a \rightarrow|$
ω
$U_{S_\omega} = \underbrace{0 \cdots 0}_{b-a}\underbrace{1 \cdots 1}_{\omega}\underbrace{0 \cdots 0}_{a-\omega}$
$L_{S_\omega} = \underbrace{0 \cdots 0}_{b-\omega}\underbrace{1 \cdots 1}_{\omega}$
$b - \omega$

Fig. 4. Parallelogram area enclosed between the lower and upper bound values for the Hamming weight class S_ω range query $[L, U] = [0, 2^a]$.

For any value $x \in [0, n-1]$, if its Hamming weight $HW(x) = \omega$ is $\omega \le a \le a'$, then x must belong to the class S_ω. Based on the above discussion, the range $[0, 2^{a'}]$ can generate a parallelogram with the length $(a' - \omega)$ times the height ω, and the range $[0, 2^a]$ can generate a parallelogram with the length $(a - \omega)$ times the same height ω in the matrix R_ω, as shown in Fig. 5. Hence, if x really belongs to the range $[L = 2^a, U = 2^{a'}]$, its cumulative sum will be within the yellow-shaded parallelogram area in the matrix R_ω. As a result, for each matrix R_ω, where $\omega \le a \le a'$, we can first find the lower bound $L_{S_\omega} : \underbrace{0 \cdots 0}_{b-a}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}}\underbrace{0 \cdots 0}_{a-\boldsymbol{\omega}}$, the upper bound $U_{S_\omega} : \underbrace{0 \cdots 0}_{b-a'}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}}\underbrace{0 \cdots 0}_{a'-\boldsymbol{\omega}}$, and fill all entries between them as ones can support the range query $[L, U] = [2^a, 2^{a'}]$.

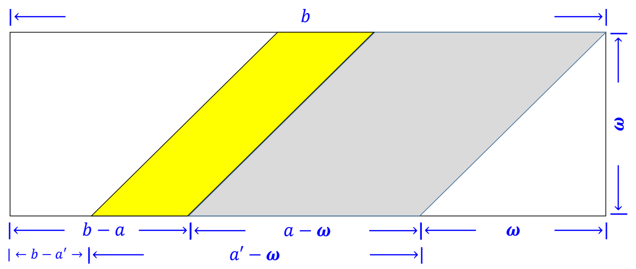


Fig. 5. Yellow-shaded parallelogram area enclosed between the lower and upper bound values for the Hamming weight class S_ω in range query $[L, U] = [2^a, 2^{a'}]$.

For any value $x \in [0, n-1]$, if its Hamming weight $HW(x) = \omega$ is $a < \omega \le a'$, then x must belong to the class S_ω. Because $\omega > a$, the range $[0, 2^a]$ does not contribute any element in class S_ω. Therefore, all elements in S_ω are generated from the range $[0, 2^{a'}]$. As a result, for each matrix R_ω, where $a < \omega \le a'$, we can first find the lower bound $L_{S_\omega} : \underbrace{0 \cdots 0}_{b-\boldsymbol{\omega}}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}}$, the upper bound $U_{S_\omega} : \underbrace{0 \cdots 0}_{b-a'}\underbrace{1 \cdots 1}_{\boldsymbol{\omega}}\underbrace{0 \cdots 0}_{a'-\boldsymbol{\omega}}$, and fill all entries between them as ones can support the range query $[L, U] = [2^a, 2^{a'}]$. By summarizing all of the above discussions, we complete the proof of Theorem 3. ■

B. Communication Overhead of the Decomposition Technique

Since the novelty of our $O(\log^3 n)$ communication-efficient range query drives its efficiency form our novel decomposition process, here we will briefly review the decomposition steps and prove its efficiency. Our decomposition technique will convert the original range query $[L, U]$ into corresponding matrices, namely semi-triangular matrices R_i's which ultimately result in sparse matrices T_i's. In the following, we will determine the exact number of ciphertexts to form the distinct semi-triangular and sparse matrices. First, both T_0 and T_b are 1×1 matrices, so the communication cost (denoted as $\mathbb{C}$) is $\mathbb{C} = 2$. Second, the remaining R_i's, $1 \le i \le b-1$ can be organized by $b-1$ matrices in which each one has b columns. Clearly, the last column, cells shaded in yellow color, can be discarded and since each matrix R_i and the corresponding T_i contain $i+1$ rows, therefore the total number of required ciphertexts in T_i is $(b-1) * (i+1)$. Up to this point, the communication cost is $\mathbb{C} = 2 + \sum_{i=1}^{b-1}(b-1)(i+1)$. Finally, the unused area of each T_i should be eliminated with a considerable reduction in the number of transmitted ciphertexts. Specifically, $1 + 2 + \cdots + (i-1) = \frac{i(i-1)}{2}$ reflects the appropriate number of shaded cells in R_i that can be eliminated in this step to form each T_i. As a result, the proposed algorithm's communication cost $\mathbb{C}$ is

$$\begin{aligned}\mathbb{C} &= 2 + \sum_{i=1}^{b-1}(b-1)(i+1) - \sum_{i=1}^{b-1}\frac{i(i-1)}{2} \\ &= 2 + \frac{(b-1)^2(b+2)}{2} - \frac{(b-1)b(b-2)}{6} \\ &= 2 + \frac{2b^3+3b^2-11b}{6} \in O(\log^3 n)\end{aligned}$$

C. Description of Our Proposed Scheme

We will now give a detailed description of our communication-efficient privacy-preserving range query scheme in fog-based IoT. It consists of five phases: 1) query user key generation, 2) range query generation at query user, 3) query response at IoT devices, 4) response aggregation at the fog node, and 5) response recovery at query user.

1) Query User Key Generation: For the simplicity's sake, we consider $n = 2^b$ to represent n as a b-bit binary value. Given the input parameters (k_0, k_1, k_2), the query user generates the secret key $SK = (p, q, \mathcal{L})$ and the public parameter PP based on our previously described SHE scheme. Then, the query user keeps the secret key SK and sends the public parameter PP to the fog node and IoT devices $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$. In order to achieve $O(\log^3 n)$ communication efficiency, the following two conditions should be satisfied:

- The message space depends on k_1 and it should be set at least to $\lceil \log_2 N \rceil$ to successfully recover the query response COUNT($\mathbf{I}'$) = $|\mathbf{I}'|$, where N is the number of IoT devices.
- The SHE scheme should accept at least b homomorphic multiplication operations, therefore k_0 should be set equal to $2(b+1)k_2$ in order to successfully evaluate the b-depth multiplicative circuit.

2) Range Query Generation at Query User: When the query user intends to request a range query $[L, U]$, where $0 \le L \le U \le n-1 = 2^b - 1$ and $L = 2^a, U = 2^{a'}$, he/she can generate the query request in the following steps.

Step 1: Transform the range $[L, U]$ to be $b+1$ matrices $\{T_0, T_1, \cdots, T_b\}$ by following the decomposition technique in Subsection IV-A.

Step 2: Encrypt each T_i for $0 \le i \le b$. In specific, in the matrix T_i, the value zero and one is encrypted to be $Enc(0)$ and $Enc(1)$ with the encryption method of our proposed SHE. Then, the query user forwards the encrypted sparse matrices $\{Enc(T_0), Enc(T_1), \cdots, Enc(T_b)\}$ to the IoT devices via fog node. At the same time, besides the encrypted T_i's, an extra $Enc(0)$ is also generated and sent to each IoT device, which can provide the self-blindness functionality for our scheme. For example, the IoT device can multiply the additional $Enc(0)$ to a random integer value, and then the result is added to encrypted prepared response to re-randomize the response for protecting against the curious fog node.

3) Query Response at IoT Devices: Upon receiving encrypted sparse matrices $\{Enc(T_0), Enc(T_1), \cdots, Enc(T_b)\}$, each IoT device I_i with prepared data x_i will perform the following steps to report its encrypted response. Note that, though the employed SHE is a symmetric encryption algorithm, when each IoT device reports its encrypted response, it does not need to know the secret key SK. Instead, each IoT device can directly perform over the ciphertexts. We will illustrate each step by using two different examples, where x_i is within the range and out of the range. Recall the example in Fig. 2 where the user range is bounded within $[8, 16], 0 \le L \le U \le n-1 = 31$ and consider two different x_i, e.g., 7 is within and 10 is out of the range.

Step 1: IoT device I_i converts its prepared data x_i into binary form, e.g., $7 = (00111)$ and $10 = (01010)$.

Step 2: Compute the Hamming weight of the binary value x_i, i.e., $\alpha = \textsc{HammingWeight}(x_i)$, which is calculated to indicate the index of sparse matrix T_i that should be used to calculate the encrypted result, e.g., $HW\Big(7 = (00111)\Big) = 3$ and $HW\Big(10 = (01010)\Big) = 2$, therefore IoT device will investigate T_3 and T_2, respectively for $x_i = 7$ and $x_i = 10$.

Step 3: The IoT device generates the cumulative sum value with respect to x_i, i.e., $\beta_{[0..b-1]} = \textsc{CummulativeSum}(x_i)$. For example, the cumulative sum for $x_i = 7$ is 00123 and in case of $x_i = 10$ is 01122.

Step 4: The IoT device I_i calculates the following multiplicative expression as the encrypted response.

$$Response_{(I_i)} = \prod_{k=0}^{b-1} T_\alpha[\beta(k)][k]$$

For example, the encrypted response in case of $x_i = 7 = (00111)_2$ is

$$\begin{aligned}&Response_{(I_i)} = \prod_{k=0}^{4} T_\alpha[\beta(k)][k] \\ &= T_3[\beta(0)][0] \times T_3[\beta(1)][1] \times \cdots T_3[\beta(4)][4] \\ &= T_3[0][0] \times T_3[0][1] \times T_3[1][2] \times T_3[2][3] \times T_3[3][4] \\ &= Enc(1) \times Enc(0) \times Enc(1) \times Enc(1) \times Enc(1) \\ &= Enc(0) \therefore 7 \notin [8, 16]\end{aligned}$$

While for $x_i = 10 = (01010)_2$, the encrypted response is

$$\begin{aligned}&Response_{(I_i)} = \prod_{k=0}^{4} T_\alpha[\beta(k)][k] \\ &= T_2[\beta(0)][0] \times T_2[\beta(1)][1] \times \cdots T_2[\beta(4)][4] \\ &= T_2[0][0] \times T_2[1][1] \times T_2[1][2] \times T_2[2][3] \times T_2[2][4] \\ &= Enc(1) \times Enc(1) \times Enc(1) \times Enc(1) \times Enc(1) \\ &= Enc(1) \therefore 10 \in [8, 16]\end{aligned}$$

Step 5: To protect $Response_{(I_i)}$ from a curious fog node, the IoT device I_i self-blinds $Response_{(I_i)}$ with the extra prepared $Enc(0)$ as

$$Response_{(I_i)} = Response_{(I_i)} + \Big(Enc(0) \times r\Big)$$

where $r \in \mathbb{Z}_n$. Finally, I_i sends the self-blinded encrypted response $Response_{(I_i)}$ to the fog node.

4) Response Aggregation at Fog Node: After receiving all self-blinded responses from IoT devices, the fog node aggregates the intermediate encrypted responses to generate the final query result as

$$Count = \sum_{I_i \in \mathbf{I}} Response_{(I_i)}$$

Then, the fog device forwards $Count$ to the user.

5) Response recovery at query user: On receiving the encrypted query result $Count$, the user uses secret key SK to recover the range query response as

$$Count \xrightarrow{Dec} \text{Count}(\mathbf{I}') = |\mathbf{I}'| = \sum_{I_i \in \mathbf{I}} Response_{(I_i)}$$

The correctness of the result is:

$$\begin{aligned} Count &= \sum_{I_i \in \mathbf{I}} Response_{(I_i)} = \sum_{I_i \in \mathbf{I}} \left(\prod_{k=0}^{b-1} T_\alpha[\beta(k)][k] \right) \\ &= \sum_{I_i \in \mathbf{I}'} Enc(1) + \sum_{I_i \notin \mathbf{I}'} Enc(0) = Enc(|\mathbf{I}'|) \end{aligned}$$

V. Security Analysis

In this section, we analyze the security of our proposed privacy-preserving range query scheme. Particularly, we focus on the privacy properties, i.e., range query $[L, U]$ and the subset $\mathbf{I}'$ should be privacy-preserving.

• *The query range $[L, U]$ is privacy-preserving in the proposed scheme.* As described in Subsection IV-C, in order to achieve the communication efficiency, the query range $[L, U]$ will be transformed into $b+1$ sparse matrices $\{T_0, T_1, \cdots, T_{b=\log_2 n}\}$. At the same time, each value in matrix T_i will be encrypted by our SHE cryptosystem, which was shown to be secure against the known-plaintext attack in Theorem 2. Then, the security of our SHE cryptosystem guarantees that without knowing the secret key $SK = (p, q, \mathcal{L})$ the adversary has no idea on the matrix T_i for $0 \leq i \leq b$. Since both the fog device and IoT devices cannot access to the secret key SK, they have no idea on the matrix T_i ($0 \leq i \leq b$), i.e., they cannot distinguish whether a ciphertext in encrypted T_i is encrypted from zero or one. Furthermore, without any plaintext information about each T_i, the fog node and IoT devices definitely have no idea on the query range $[L, U]$. Therefore, the query range $[L, U]$ in the proposed scheme can be kept secret from both the fog node and IoT devices, as a result it is privacy-preserving.

• *The subset $\mathbf{I}'$ is also privacy-preserving in the proposed scheme.* In the proposed scheme, the subset $\mathbf{I}'$ denotes a set of IoT devices whose data are within the query range $[L, U]$, i.e., $\mathbf{I}' = \{I_i | I_i \in \mathbf{I} \ \wedge x_i \in [L, U]\}$. As described in our security model, $\mathbf{I}'$ should be kept secret from the query user, fog device, and IoT devices. For each IoT device I_i, if it attempts to obtain the information about $\mathbf{I}'$, it needs to determine whether itself is in $\mathbf{I}'$ or not and whether other IoT devices are in $\mathbf{I}'$ or not. For I_i, it uses its prepared data x_i to compute the encrypted $Response_{(I_i)}$, but has no idea on $Response_{(I_i)}$ is $Enc(0)$ or $Enc(1)$. In other words, I_i does not know whether w_i is in the query range $[L, U]$ or if I_i is in $\mathbf{I}'$. For the other IoT devices, since I_i cannot access the plaintext data prepared by them and the plaintext of query range, it also has no idea on whether other IoT devices are in $\mathbf{I}'$ or not. Thus, the subset $\mathbf{I}'$ can be kept secret from the IoT devices.

For the fog node, it receives the encrypted query range $[L, U]$ from the user and the encrypted response $Response_{(I_i)}$ from each IoT device I_i. It is likely to use them to deduce the information of the subset $\mathbf{I}'$. However, on the one hand, the query range $[L, U]$ is transformed into matrices $\{T_0, T_1, \cdots, T_b\}$ and are encrypted by our SHE cryptosystem. As analyzed above, our SHE cryptosystem guarantees that the fog node has no idea of the plaintext of the query range. On the other hand, each IoT device's encrypted response $Response_{(I_i)}$ is also encrypted by our SHE cryptosystem. At the same time, the $Response_{(I_i)}$ is self-blinding with the factor $\Big(Enc(0) \times r\Big)$. Note that, if there is no self-blinding factor $\Big(Enc(0) \times r\Big)$, it is possible for the fog node to identify the specific IoT device's response. Specifically, it can iterate over all possible prepared values x_i ($0 \leq x_i \leq n-1$) by generating Hamming weight and cumulative sum values as well as multiplying the corresponding sparse matrix entries to check whether $Response_{(I_i)} \overset{?}{=} \prod_{k=0}^{b-1} T_\alpha[\beta(k)][k]$. Although it requires a lot of computations, it is feasible, especially when n is small value. Luckily, $Response_{(I_i)} = Response_{(I_i)} + \Big(Enc(0) \times r\Big)$ includes the random factor $\Big(Enc(0) \times r\Big)$, which makes it impossible for the fog node to launch a brute force attack on IoT responses. In this case, it is almost impossible for the fog node to obtain the information about the subset $\mathbf{I}'$. Thus, the subset $\mathbf{I}'$ can be kept secret from the fog node.

For the query user, he/she receives the encrypted query response from the fog node and recovers the plaintext of the query result by decryption. As described in Subsection IV-C, the query response is an aggregated result, i.e., $|\mathbf{I}'|$. Thus, the query user only knows the number of IoT devices whose data are within the query range, but has no idea which specific IoT device has a data within the query range. Thus, the subset $\mathbf{I}'$ can be kept secret from the query user under our considered security model.

VI. Performance Evaluation

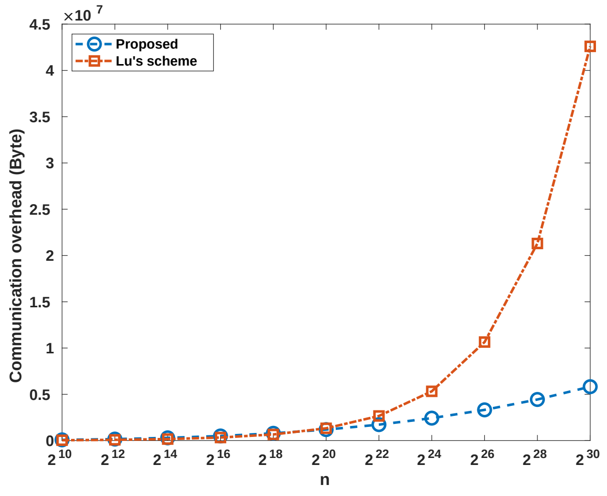
In this section, we evaluate the performance of our proposed scheme with respect to the communication overhead and computational cost. Specifically, we compare our proposed scheme with a recently proposed privacy-preserving range query scheme, i.e. Lu's scheme [16]. Lu's privacy-preserving range query scheme with $O(\sqrt{n})$ communication complexity is performed based on BGN homomorphic encryption [17]. In our performance evaluation, we implemented both our scheme and Lu's scheme in Java (JDK 1.8 update 222) on a machine with Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz, 8 GB main memory, and Linux (Ubuntu 16.04) operating system. The detailed parameter settings for both schemes are listed in Table II. We repeated each experiment 10 times and report the average results.

TABLE II
THE PARAMETER SETTINGS

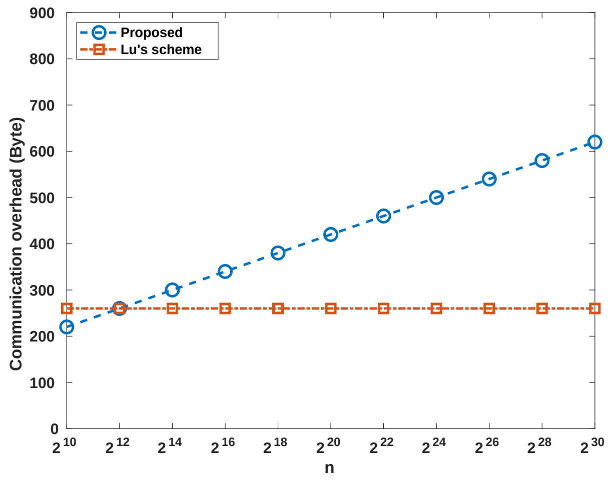
Shared parameter	Value
N	The number of IoT devices, $N = 1000$
n	The upper bound of $[1, n]$, $n = \{2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}, 2^{20}, 2^{22}, 2^{24}, 2^{26}, 2^{28}, 2^{30}\}$
$[L, U]$	The range query lower and upper bound, $0 \leq L \leq U \leq n - 1$
x_i	IoT device D_i's prepared data, randomly picked from $[1, n]$ where $0 \leq w_i \leq n - 1$

Proposed scheme based on SHE	
Parameter	Value
b	$b = \log_2(n)$, $b = \{10, 12, 14, \cdots, 30\}$
k_0, k_1, k_2	$\lvert k_2 \rvert = 40$, $k_1 = \log_2 N$, $k_0 = 2(b+1)k_2$
p, q ,$\mathcal{L}$, $\mathcal{N}$	$\lvert p \rvert = \lvert q \rvert = k_0$, $\lvert \mathcal{L} \rvert = k_2$, $\mathcal{N} = pq$
$\mathcal{M}$	The query result $\mathcal{M} = N \in \{0, 1\}^{k_1}$

Lu's scheme based on BGN Homomorphic Encryption	
Parameter	Value
b	$b = \sqrt{n}$, $b = \{2^5, 2^6, 2^7, \cdots, 2^{15}\}$
κ	$\kappa = 512$
p, q, $\mathcal{N}$	$\lvert p \rvert = \lvert q \rvert = \kappa = 512$, $\mathcal{N} = pq$
Δ	The upper bound of query result, $\Delta = N$



(a) Request from user to fog node



(b) Response from fog node to user

Fig. 6. The communication overhead comparison between our proposed scheme and Lu's scheme with respect to n varying from 2^{10} to 2^{30}

A. *Communication Overhead*

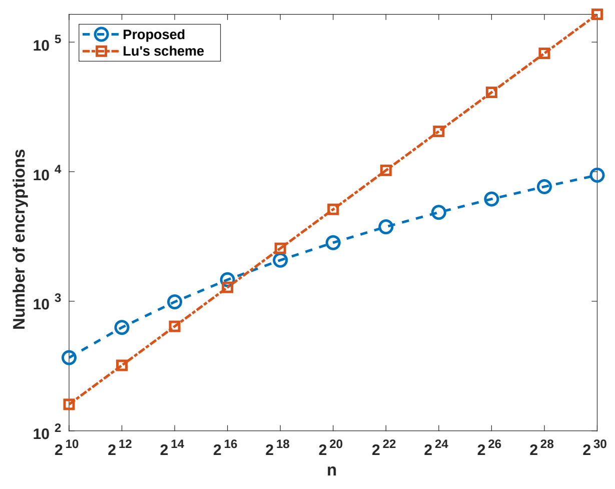
In this subsection, we evaluate the communication overhead of both the query request and the query response. First, Fig. 6(a) plots the communication overhead of transferring an encrypted query request from user to fog node in both schemes with n varying from 2^{10} to 2^{30}. From the figure, we see the communication overhead of Lu's scheme is growing exponentially, while ours remains very low and highly efficient. This is because Lu's scheme has to upload $O(\sqrt{n})$ encrypted data from the user to the fog node, while ours only uploads $O(\log^3 n)$ encrypted data. Second, Fig. 6(b) shows the communication overhead of a query response from the fog node to the user in both schemes with respect to n. One can see that both schemes are communication efficient. Although the response cost in our proposed scheme increases with the parameter n, but it is acceptable and as low as almost 600 bytes for $n = 2^{30}$.

B. *Computational Cost*

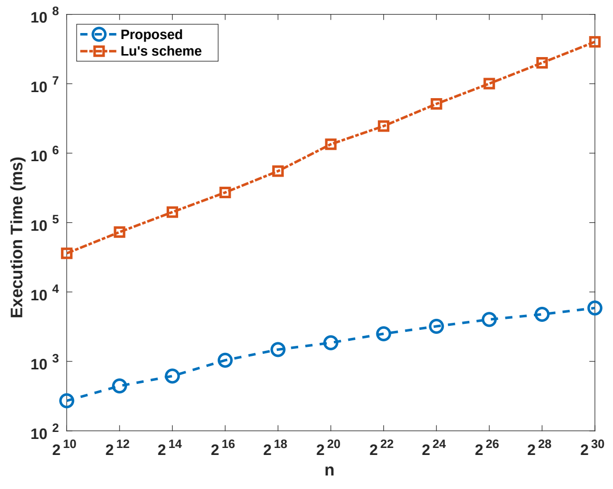
The computational cost, of both schemes, is influenced by the number of encryptions performed by the user during the range query formulation and decomposition phase. We first compare the number of encryption operations in both schemes with respect to n as shown in Fig. 7(a). From this figure, we can see that the number of encryptions in Lu's scheme is rapidly increasing with n. This is because of the difference in underlying decomposition technique. Lu's scheme uses $O(\sqrt{n})$ decomposition technique, whereas our scheme uses an $O(\log^3 n)$ one.

We now evaluate the computational cost of both schemes at each step of range query processing with respect to n as shown in Fig. 7. Specifically, Fig. 7(b) shows the query generation execution time, which indicates that the computational cost of Lu's scheme grows exponentially with n, while ours grows much more slowly due to ours having fewer encryptions. Fig. 7(c) illustrates the query response time of both schemes for different problem sizes. Although, the processing time in Lu's scheme is constant, it is still considerably higher than ours (only one millisecond for $n = 2^{30}$). This is mainly because of the time-consuming pairing operations and costly homomorphic calculations in BGN compared with fast and efficient multiplicative homomorphic operations in our SHE scheme.

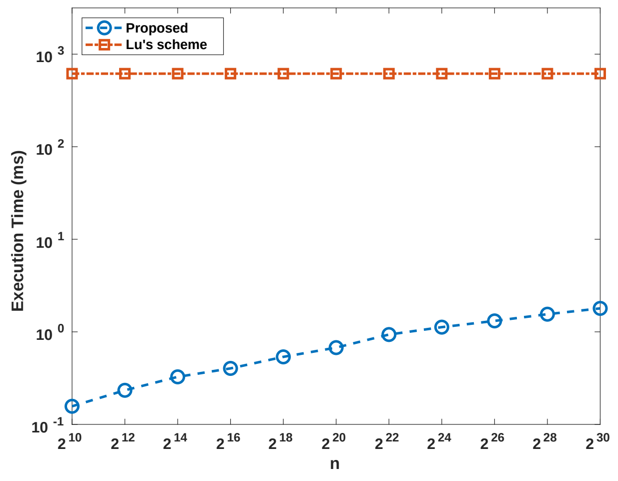
Fig. 7(d) depicts the response aggregation time at fog node. Merging the IoT devices' responses at fog node is mainly affected by the number of IoT devices, i.e. N. Therefore, as it is apparent from figure, the aggregation time in both schemes is independent of problem size, even if there are small increments in our proposed scheme, varying with n.



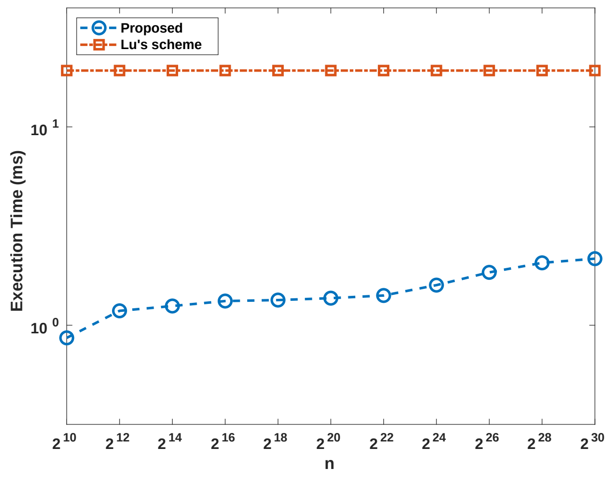
(a) The number of encryptions in the decomposition phase



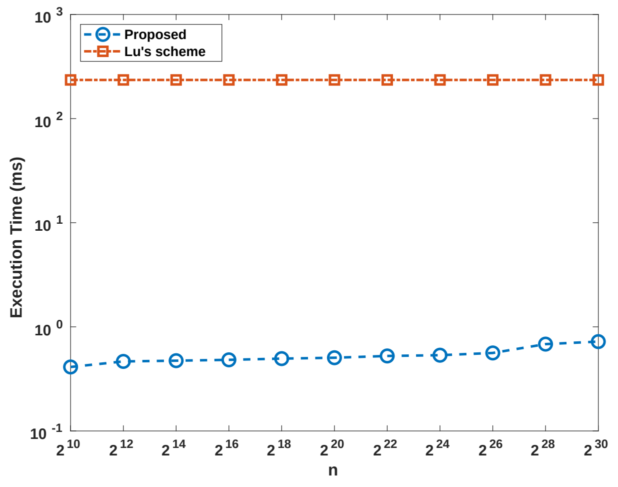
(b) Query generation at the user side



(c) Query response at the IoT device



(d) Response aggregation at the fog node



(e) Response recovery at the user side

Fig. 7. The computational cost comparison between our scheme and Lu's scheme with respect to n varying from 2^{10} to 2^{30}

Our scheme has insignificant changes with respect to n due to gradually increasing ciphertext lengths, while in Lu's scheme the cipthertext length only depends on κ value and is consequently independent of n.

Finally, Fig. 7(e) presents the response recovery time which measures how long time the user spent decrypting the ciphertext upon receiving the encrypted result from fog node. Obviously, they are still efficient and independent of problem size n. However, decryption in our proposed scheme is faster than Lu's. Because Lu's scheme performs in BGN and it highly depends on the message space Δ, it can only operate effectively with relatively smaller Δ.

VII. Related Work

Recently, some related works have emerged on privacy-preserving range queries in the fog-based IoT [16], which are closely related to our proposed query scheme. As discussed in Section I, in order to achieve the desired security properties for a privacy-preserving range query, the naive solution has an $O(n)$ communication cost for the range query. For a toy example with $n = 10$, the naive solution is efficient enough. However, when n becomes large, the efficiency, especially the communication efficiency rapidly deteriorates. To improve this, Lu [16] proposed a new privacy-preserving range query scheme built upon BGN homomorphic encryption together with a novel range query expression, decomposition, and composition technique. It had an $O(\sqrt{n})$ communication efficiency. Therefore, when n becomes larger, Lu's scheme is much more efficient than the naive solution in terms of communication overhead. However, when n becomes even larger, e.g., $n = 2^{30}$, the $O(\sqrt{n})$ communication cost seems too costly. In addition, because Lu's scheme is based on the BGN with time-consuming bilinear pairing operations in public key settings, the computational cost is also expensive. Aiming to improve the communication efficiency, in this paper, we proposed a more efficient privacy-preserving range query scheme in the fog-based IoT. Using a novel range query decomposition technique and an efficient symmetric homomorphic scheme (SHE), our proposed scheme achieves $O(\log^3 n)$ communication-efficiency.

When the privacy-preserving range query on "*the number of IoT devices whose data x_i is within the range [L,U]*" is considered as a special privacy-preserving data aggregation scheme in fog-based IoT, there are other related works close to our proposed scheme. In [21], Lu et al. addressed heterogeneous data aggregation in real IoT applications by proposing a light-weight privacy-preserving data aggregation (LDPA) for fog-enabled setting. The proposed LPDA is characterized by applying Paillier cryptosystem, Chinese Remainder Theorem, and a one-way hash chain function to aggregate hybrid IoT devices' data and to early filter the injected false data at the network edge. In [22], Huang et al. studied the fog-assisted selective aggregation operation. Specifically, they constructed a new threat model to formalize the non-collusive and collusive attacks of compromised fog nodes, and their proposed privacy-preserving and reliable selective multi-source aggregation scheme, which is comprised of BCP cryptosystem, randomized message-lock encryption, homomorphic proxy-authenticators, and multi-dimensional aggregation, can well tackle the data privacy and reliability challenges. Most recently, Mahdikhani et al. [23] presented a privacy-preserving subset aggregation scheme in fog-enhanced IoT scenarios, which enables a query user to gain the sum of the prepared data from a subset of IoT devices. To identify the subset, the inner product similarity of the normalized vectors in the query user side and each IoT device is securely computed. Only when the inner product is greater than the user's specified threshold will an IoT device's data be privately aggregated to form the final response.

VIII. Conclusion

In this paper, we have proposed a communication-efficient privacy-preserving range query scheme for fog-based IoT. For the range queries of the form $[L = 2^a, U = 2^{a'}]$, the proposed scheme is characterized by designing a novel range decomposition technique to reduce the communication overhead to be $O(\log^3 n)$, and devising an efficient homomorphic encryption scheme to preserve the data privacy for both the range query and individual IoT device's data. To the best of our knowledge, the communication overhead of our proposed scheme is much lower than that of the most communication efficient privacy-preserving range query scheme, i.e., Lu's scheme, whose communication overhead is $O(\sqrt{n})$. Detailed security analysis shows that our scheme is indeed privacy-preserving under our defined security model. An extensive performance evaluation validated the efficiency of our scheme in terms of communication overhead and computational cost. In our future work, we plan to expand this study by: 1) investigating more general privacy-preserving range query functions, e.g., multi-range query, and 2) launching more complex configuration in the system model.

Acknowledgement

This work is supported in part by NSERC Discovery Grants (no. Rgpin 04009), Natural Science Foundation of Zhejiang Province (grant no. LZ18F020003), and National Natural Science Foundation of China (grant no. U1709217).

References

[1] W. A. Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary, and K. Akkaya, "Privacy-preserving smart parking system using blockchain and private information retrieval," *arXiv preprint arXiv:1904.09703*, 2019.

[2] M. Baza, N. Lasla, M. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," *IEEE Transactions on Network Science and Engineering*, 2019.

[3] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *IEEE Communications magazine*, vol. 49, no. 11, pp. 68–75, 2011.

[4] X. Lin, R. Lu, and X. Shen, "Mdpa: multidimensional privacy-preserving aggregation scheme for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 6, pp. 843–856, 2010.

[5] B. Shabandri and P. Maheshwari, "Enhancing iot security and privacy using distributed ledgers with iota and the tangle," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2019, pp. 1069–1075.

[6] EETimes, "5 ways the food industry can improve food safety with the iot," https://iot.eetimes.com/5-ways-the-food-industry-can-improve-food-safety-with-the-iot/, 2019.

[7] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "Parq: A privacy-preserving range query scheme over encrypted metering data for smart grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 178–191, 2013.

[8] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "Muda: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-peer networking and applications*, vol. 8, no. 5, pp. 777–792, 2015.

[9] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet of Things Journal*, vol. 2, no. 3, pp. 248–258, 2015.

[10] TechTarget, "The impact of iot on big data," https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/The-impact-of-IoT-on-big-data, 2019.

[11] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[12] S. P. Singh, A. Nayyar, R. Kumar, and A. Sharma, "Fog computing: from architecture to edge computing and big data processing," *The Journal of Supercomputing*, pp. 1–36, 2018.

[13] Q. Kong, R. Lu, M. Ma, and H. Bao, "A privacy-preserving and verifiable querying scheme in vehicular fog data dissemination," *IEEE Trans. Vehicular Technology*, vol. 68, no. 2, pp. 1877–1887, 2019.

[14] H. Mahdikhani and R. Lu, "Achieving privacy-preserving multi dot-product query in fog computing-enhanced iot," in *2017 IEEE Global Communications Conference, GLOBECOM 2017, Singapore, December 4-8, 2017*. IEEE, 2017, pp. 1–6.

[15] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 46–52, 2018.

[16] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced iot," *IEEE Internet of Things Journal*, 2018.

[17] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, 2005, pp. 325–341.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, 1999, pp. 223–238.

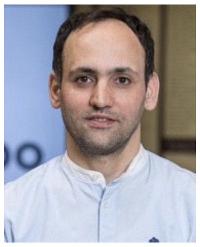
[19] Q. Wang, J. Huang, Y. Chen, C. Wang, F. Xiao, and X. Luo, "$prost$: Privacy-preserving and truthful online double auction for spectrum allocation," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 374–386, 2018.

[20] H.-T. Wu, Y.-m. Cheung, Z. Yang, and S. Tang, "A high-capacity reversible data hiding method for homomorphic encrypted images," *Journal of Visual Communication and Image Representation*, vol. 62, pp. 87–96, 2019.

[21] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[22] C. Huang, D. Liu, J. Ni, R. Lu, and X. Shen, "Reliable and privacy-preserving selective data aggregation for fog-based iot," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[23] H. Mahdikhani, S. Mahdavifar, R. Lu, H. Zhu, and A. A. Ghorbani, "Achieving privacy-preserving subset aggregation in fog-enhanced iot," *IEEE Access*, pp. 1–1, 2019.

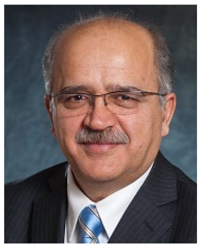


Hassan Mahdikhani holds the B.Eng.degree in Computer Engineering-Software from Kharazmi University, Tehran, Iran, in 2001 and the M.Eng. degree in Computer Engineering-Software from Iran University of Science and Technology, Tehran, Iran, in 2006. Hassan is currently a Ph.D. candidate in Computer Science at the University of New Brunswick (UNB), and a cybersecurity researcher at the Canadian Institute for Cybersecurity (CIC), Canada. Hassan's research interests include cloud computing securtiy, secure and privacy-preserving computation offloading and applied cryptography.

Rongxing Lu (S'09-M'11-SM'15) is an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor General's Gold Medal", when he received his PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently a senior member of IEEE Communications Society. Dr. Lu currently serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC. Dr.Lu is the Winner of 2016-17 Excellence in Teaching Award, FCS, UNB.

Yandong Zheng received her M.S. degree from the Department of Computer Science, Beihang University, China, in 2017 and She is currently pursuing her Ph.D. degree in the Faculty of Computer Science, University of New Brunswick, Canada. Her research interest includes cloud computing security, big data privacy and applied privacy.

Jun Shao received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2008. He was a Postdoctoral Fellow with the School of Information Sciences and Technology, Pennsylvania State University, State College, PA, USA, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His current research interests include network security and applied cryptography.



Ali A. Ghorbani (SM'-) has held a variety of positions in academic for the past 37 years is currently a Professor of Computer Science. Tier 1 Canadian Institute for Cybersecurity, the Director of the Canadian Institute for Cybersecurity, which he established in 2016, and an IBM Canada Faculty Fellow. Dr.Ghorbani is also the founding director of the laboratory for intelligence and adaptive systems research. His current research focus is Cybersecurity, WebIntelligence, and Critical Infrastructure Protection. Dr. Ghorbani is the co-inventor on 3 awarded patents in the area of Network Security and Web Intelligence and has published over 270 peer-reviewed articles during his career. He has supervised over 180 research associates, postdoctoral fellows, graduate and undergraduate students during his career. He is the co-founder of the Privacy, Security, Trust (PST) Network in Canada and its international annual conference. Dr.Ghorbani served as the co-Editor-In-Chief of Computational Intelligence: An International Journal from 2007 to 2017.