Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8139813, 17 pages
https://doi.org/10.1155/2022/8139813

Research Article

Differential Privacy via Haar Wavelet Transform and Gaussian Mechanism for Range Query

Dong Chen, Yanjuan Li, Jiaquan Chen, Hongbo Bi, and Xiajun Ding

College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China

Correspondence should be addressed to Xiajun Ding; 37050@qzc.edu.cn

Received 13 May 2022; Revised 3 July 2022; Accepted 22 August 2022; Published 12 September 2022

Academic Editor: Lorenzo Putzu

Range query is the hot topic of the privacy-preserving data publishing. To preserve privacy, the large range query means more accumulate noise will be injected into the input data. This study presents a research on differential privacy for range query via Haar wavelet transform and Gaussian mechanism. First, the noise injected into the input data via Laplace mechanism is analyzed, and we conclude that it is difficult to judge the level of privacy protection based on the Haar wavelet transform and Laplace mechanism for range query because the sum of independent random Laplace variables is not a variable of a Laplace distribution. Second, the method of injecting noise into Haar wavelet coefficients via Gaussian mechanism is proposed in this study. Finally, the maximum variance for any range query under the framework of Haar wavelet transform and Gaussian mechanism is given. The analysis shows that using Haar wavelet transform and Gaussian mechanism, we can preserve the differential privacy for each input data and any range query, and the variance of noise is far less than that just using the Gaussian mechanism. In an experimental study on the dataset age extracted from IPUM's census data of the United States, we confirm that the proposed mechanism has much smaller maximum variance of noises than the Gaussian mechanism for range-count queries.

1. Introduction

Over the past ten years, differential privacy has become one of the important methods in the area of privacy-preserving for statistical databases. Differential privacy is a promising scheme for publishing statistical query results of sensitive data, which has a strong privacy guarantee for opponents with arbitrary background knowledge [1–6]. The strong privacy guarantee of differential privacy ensures that any individuals in the data set will not significantly affect the analysis results of the data set. At present, three basic mechanisms are widely used to ensure differential privacy: Laplace mechanism, Gaussian mechanism, and exponential mechanism. Laplacian and Gaussian mechanisms are applicable to numerical queries, and exponential mechanisms are applicable to non-numeric queries [7–9]. Recently, differential privacy is adopted on many research field, such as social network publishing [10–12], crowdsourced data publication [13, 14], and genomic privacy [15–17].

Along with a long-range query scope, the accumulation of noise in the range query answered for privacy preserving can affect the usability of the released data [18, 19]. To reduce the accumulation of noise, the method of hierarchical decompositions is usually employed [20]. Zhang et al. proposed a differentially private algorithm for hierarchical decompositions and named it as PrivTree. This histogram construction algorithm eliminates the dependency on a predefined limit parameter. The privacy-preserving range query is adopted in the field of Internet of Things (IoT) in recent years [21–23]. Cai et al. studied the transaction approximate range counting problem of large IoT data. They proposed a sampling-based method to generate approximate counting results. For privacy reasons, these results will be further disturbed and then published. It is theoretically proved that this result achieves unbiasedness, bounded variance and enhances privacy guarantee under differential privacy. Mahdikhani et al. proposed a communication efficient privacy protection range query in the fog-enhanced Internet of things. The feature of this scheme is that it adopts the Paillier homomorphic cryptosystem and the ingenious bloom filter data structure to achieve better privacy and higher count aggregation efficiency in the range query scenario of protecting privacy. Histogram is a representative and popular tool for data publishing and visualization tasks. Nowadays, protecting private data and preventing the leakage of sensitive information have become one of the main challenges faced by histogram [24–26]. Histogram is the result of a group of counting queries. It is the core statistical tool for reporting data distribution. In fact, it is regarded as the basic method of many other statistical analyses, such as range query [27]. The advantage of histogram representation is that it limits the sensitivity to noise. For example, when histograms are used to support range or count queries, adding or deleting a single record will affect at most one box. Therefore, the sensitivity of range or count query on the histogram is equal to 1, and the amount of additional noise per box will be relatively small [28]. For the differential privacy of long-range queries on the histogram, the accumulation of noise is a key issue that needs to be focused.

Discrete wavelet transform (DWT) is an important technology in signal and image processing [29–31]. Lifting scheme, also called second generation wavelet, has many advantages comparing with the first generation wavelet, such as in-place computation, integer-to-integer transforms, and speed [32–34]. Wavelet-based privacy preserving is studied in recent years [35–37]. Xiao et al. propose the differential privacy via Haar wavelet transform. They introduce a data publishing technique named Privelet. Privelet not only ensures ε-differential privacy but also provides accurate results for range query by injecting less noise into wavelet coefficients. The mechanism that can be used to build the differential privacy in Privelet is Laplace distribution. The Laplace mechanism, which is used to guarantee differential privacy in Privelet, maybe not a good choice for building the privacy-preserving system based on discrete wavelet. The reason is that the Laplace noise does not have the property of additivity. That is, the sum of two Laplace distributions is not a Laplace distribution. That means we cannot obtain an analyzable noise distribution by wavelet reconstruction where the Laplace noise is injected into the wavelet coefficients.

The Gaussian mechanism for differential privacy is proposed by Dwork [38, 39]. The Gaussian noise can be used in the structuring of hierarchical decompositions, such as wavelet transforms. The property of additivity of Gaussian noise is very important for the reconstruction of noise data. On the one hand, additivity can ensure that the reconstructed noise is still Gaussian noise; on the other hand, some noise can be eliminated during reconstruction.

In view of the above analysis, we will do some research on differential privacy via Gaussian mechanism and lifting scheme of Haar wavelet transform for range query in this study. In summary, the main contributions of this work are as follows:

(1) Differential privacy using lifting Haar wavelet transform and Laplace mechanism is analyzed in this study. The distribution of noise injected into the input data via wavelet reconstruction is discussed and we conclude that they are not noise of Laplace distribution.

(2) Differential privacy based on lifting Haar wavelet transform and Gaussian mechanism is constructed in this study. For range query, our analysis shows that the noise actually added into a certain range of original data is much less than the sum of noise at each data for the proposed mechanism.

(3) Differential privacy for range query via lifting Haar wavelet and Guassian mechanism is discussed. We give an algorithm to compute the maximum variance of any range query for any given parameter l (suppose the length of input data is 2^l). Moreover, we give a coarse estimation of the maximum variance of range query using a function expression. Finally, we give an experimental study using the proposed mechanism, and the results show the proposed mechanism has a much smaller maximum variance of noise than the Gaussian mechanism for range query.

The remainder of the study is organized as follows: Section 2 introduces the fundamental definitions and theorems about the differential privacy and its implement mechanism. Section 3 gives the theorems for how to inject Gaussian noise into the Haar wavelet coefficients. Section 4 analysis the noise of range query under the framework of Gaussian mechanism and Haar wavelet. First, the computing method for the variance of range query is given. Second, the algorithm of computing maximum variance for any range query is introduced, and how to get the interval of the range query when obtaining the maximum variance is introduced in detail. Finally, the coarse estimation of the maximum variance of range query is given as a function expression. Section 5 introduces the experimental verification of the computing of maximum variance for the range query based on Gaussian mechanism and lifting Haar wavelet. Conclusions is given in Section 6.

2. Preliminaries

In this section, the fundamental definitions and theorems about the differential privacy and its implement mechanism are introduced first. Furthermore, the method of injecting Laplace noise into the Haar wavelet coefficients is given. They are the basis of the other sections.

2.1. Differential Privacy

Definition 1 ((ε, δ)-Differential privacy [38, 39]). A randomized mechanism M with domain $\mathbb{N}^{|\chi|} \longrightarrow \mathbb{R}^d$ is (ε, δ)-differential privacy if for all $S \subseteq Range(M)$ and for all $x, y \in \mathbb{N}^{|\chi|}$ such that $\|x - y\|_1 \leq 1$.

$$\Pr[M(x) \in S] \leq \exp(\varepsilon)\Pr[M(y) \in S] + \delta, \quad (1)$$

where the symbol $\|x\|_1$ denotes the ℓ_1-norm of a database x, $\|x\|_1 = \sum |x_i|$, and $\|x - y\|_1$ denotes the ℓ_1-distance between two databases x and y.

2.2. Laplace Mechanism

Definition 2 (ℓ_1-sensitivity [39]). Let $\mathbb{N}^{|\chi|} \longrightarrow \mathbb{R}^d$ be an arbitrary d-dimensional function, then define the ℓ_1-sensitivity of function f as follows:

$$\begin{aligned}\Delta_1 f &= \max_{x,\, y\in\mathbb{N}^{|\chi|}} \quad \|f(x) - f(y)\|_1.\\ \|x-y\|_1 &= 1\end{aligned} \tag{2}$$

Definition 3. (Laplace distribution, $Lap(\lambda)$). The Laplace distribution with mean zero and scale λ is the distribution with probability density function:

$$Lap(x|\lambda) = \left(\frac{1}{2\lambda}\right)\exp\left(-\frac{|x|}{\lambda}\right). \tag{3}$$

In Definition 3, the variance of this distribution is $\sigma^2 = 2\lambda^2$. We write $Lap(\lambda)$ to denote the Laplace distribution with mean zero and scale λ in this study.

Theorem 1 (Laplace mechanism [39]). *Let f is a function with ℓ_1-sensitivity, the Laplace mechanism, which adds independently random drawn noise distributed as $Lap(\Delta_1 f/\varepsilon)$ into each of the d components of the output, preserves $(\varepsilon, 0)$-differential privacy.*

Remark 1. Throughout the study, we use the term "noise" to refer to a random variable with a zero mean.

2.3. Gaussian Mechanism

Definition 4 (ℓ_2-sensitivity [39]). Let $f\colon \mathbb{N}^{|\chi|} \longrightarrow \mathbb{R}^d$ be an arbitrary d-dimensional function, then define the ℓ_2-sensitivity of function f as follows:

$$\begin{aligned}\Delta_2 f &= \max_{x,\, y\in\mathbb{N}^{|\chi|}} \qquad \|f(x) - f(y)\|_2,\\ \|x-y\|_2 &= 1\end{aligned} \tag{4}$$

where the symbol $\|x\|_2$ denotes the ℓ_2-norm of a database x, $\|x\|_2 = \sum x_i^2$, and $\|f(x) - f(y)\|_2$ denotes the ℓ_2-distance between $f(x)$ and $f(y)$.

Definition 5 (Gaussian distribution, Gauss(σ^2)). The Gaussian distribution with mean zero and variance σ^2 is the distribution with probability density function:

$$Gauss\left(x|\sigma^2\right) = \frac{1}{(\sqrt{2\pi}\,\sigma)\exp\left(-x^2/\left(2\sigma^2\right)\right)}. \tag{5}$$

In Definition 5, the variance of this distribution is σ^2. We write Gauss (σ^2) to denote the Gaussian distribution with mean zero and variance σ^2.

Theorem 2 (Gaussian mechanism [38, 39]). *Let f be a function with ℓ_2-sensitivity and let $\varepsilon \in (0,1)$ be arbitrary. For $\sigma \geq \Delta_2 f \cdot \sqrt{2\ln(1.25/\delta)}/\varepsilon$, the Gaussian mechanism, which adds independently drawn random noise distributed as $Gauss(\sigma^2)$ into each of the d components of the output, ensures (ε, δ)-differential privacy.*

In Theorem 2, to ensure (ε, δ)-differential privacy, we can inject the Gaussian noise with $\sigma^2 = 2\ln(1.25/\delta)\cdot(\Delta_2 f/\varepsilon)^2$ into the input data directly.

2.4. Injecting Noise into the Input Data via Lifting Haar Wavelet

2.4.1. Lifting Scheme of Haar Wavelet. The lifting scheme of Haar wavelet transform is shown in Figure 1. In Figure 1, $x(z)$ is the input data, $x_o(z)$ and $x_e(z)$ denote the odd indexed samples and even indexed samples, respectively. $a(z)$ and $d(z)$ are the approximate coefficients and detail coefficients, respectively. For lifting scheme of Haar wavelet, we have $p(z) = -1$ and $u(z) = 1/2$.

In Figure 1, we have

$$\begin{aligned} x(z) &= \sum_{i=0}^{n-1} x_i \cdot z^{-i}\\ &= \sum x_{2k}\cdot\left(z^2\right)^{-k} + z^{-1}\cdot\sum x_{2k+1}\cdot\left(z^2\right)^{-k}\\ &= x_e\left(z^2\right) + z^{-1}x_o\left(z^2\right). \end{aligned} \tag{6}$$

Therefore, the approximate coefficients $a(z)$ and detail coefficients $d(z)$ can be given as follows:

$$a(z) = \frac{1}{2}\left(x_e(z) + x_o(z)\right), \tag{7}$$

$$d(z) = \frac{1}{2}\left(x_e(z) - x_o(z)\right). \tag{8}$$

In Figure 1, the lifting structure has the reconstruction property, that is

$$x_o'(z) = x_o(z), x_e'(z) = x_e(z), \widehat{x}(z) = x(z). \tag{9}$$

Figure 1 shows one-level decomposition and reconstruction via lifting Haar wavelet transform. The wavelet transform usually consists of many decomposition levels. We can apply the same procedure to the approximate coefficients $a(z)$ to get the multilevel Haar wavelet decomposition, as shown in Figure 2.

In Figure 2, the top decomposition level is 3 ($l = 3$), $c_{3,0}$ is the approximate coefficient, $c_{k,i}$ $(i \neq 0)$ denotes the ith wavelet coefficient in kth decomposition level, and $x_m (m \in [0,7])$ denotes the input data. In Figure 2, we observe that the number of wavelet coefficients in kth decomposition level is 2^{l-k}.

In Figure 2, given the Haar wavelet coefficients, any entry x_m can be easily reconstructed as follows:

$$x_m = c_{l,0} + \sum_{c_{k,i}\in C\setminus\{c_{l,0}\}} \left(c_{k,i}\cdot g_{k,i}\right), \tag{10}$$

where $c_{l,0}$ is the approximate coefficient, $c_{k,i} (i \neq 0)$ denotes the ith wavelet coefficient in kth decomposition level, and $g_{k,i}$ equals 1 (−1) if x_m is in the left (right) subtree of $c_{k,i}$, equals 0 if x_m is not in any subtree of $c_{k,i}$. For example,

$$x_1 = 3 = c_{3,0} + c_{3,1} + c_{2,1} - c_{1,1}. \tag{11}$$

In Figure 2, if we inject the noise into the approximate coefficients and detail coefficients, then we can obtain the reconstruction data with noise.

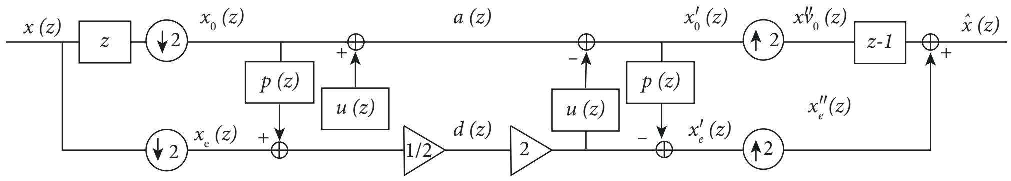


Figure 1: Lifting scheme of Haar wavelet transform.

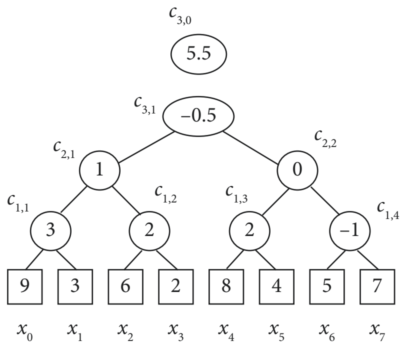


Figure 2: Multilevel lifting Haar wavelet transform.

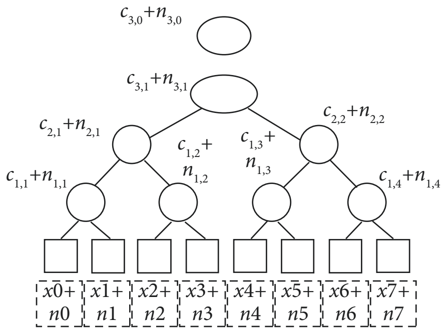


Figure 3: Getting input data with noise.

2.4.2. Injecting Noise into Haar Wavelet Coefficients. For the noise injected into Haar wavelet coefficients, the tree structure of noise can be obtained by changing the symbol "*c*" and "*x*" to *n* in Figure 2 because they use the same decomposition of multilevel lifting Haar wavelet transform.

Referring to equation (10), the noise n_m that injected into data x_m can be given as follows:

$$n_m = n_{l,0} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(n_{k,i} \cdot g_{k,i}\right), \tag{12}$$

where $n_{l,0}$ is the noise injected into approximate coefficient, $n_{k,i}$ $(i \neq 0)$ denotes the noise injected into the *i*th wavelet coefficient in *k*th decomposition level, and $g_{k,i}$ equals 1 (−1) if n_m is in the left (right) subtree of $n_{k,i}$ and equals 0 if n_m is not in any subtree of $n_{k,i}$.

The range sum of these noise has a special property; that is, some subnoise items can be eliminated when computing some sum of range count. For example, referring to Figure 2 and equation (12), we have

$$\sum_{i=0}^{2^l-1} n_i = \sum_{i=0}^{7} n_i = 8 \cdot n_{3,0}. \tag{13}$$

In the above equation, the other subnoise items except $n_{3,0}$ have been eliminated. This gives us the inspiration to apply this property to range query for differential privacy.

2.4.3. Getting Input Data with Noise. Based on the above two sections, we reconstruct the input data with noise by using the multilevel lifting Haar wavelet transform. Considering Figure 2, the input data with noise is shown in Figure 3.

In Figure 3, $x_m (m \in [0, 7])$ denotes the input data reconstructed, n_m is the noise injected into data x_m. $x_m + n_m$ denotes the input data with noise. Referring to equations (10) and (12), we have

$$\begin{aligned} x_m + n_m &= c_{l,0} + \sum_{c_{k,i} \in C \setminus \{c_{l,0}\}} \left(c_{k,i} \cdot g_{k,i}\right) + n_{l,0} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(n_{k,i} \cdot g_{k,i}\right) \\ &= \left(c_{l,0} + n_{l,0}\right) + \sum_{\substack{c_{k,i} \in C \setminus \{c_{l,0}\} \\ n_{k,i} \in N \setminus \{n_{l,0}\}}} \left(\left(c_{k,i} + n_{k,i}\right) \cdot g_{k,i}\right), \end{aligned} \tag{14}$$

where the meanings of the symbols $c_{l,0}$, $n_{l,0}$, $c_{k,i}$, $n_{k,i}$, and $g_{k,i}$ are as stated before.

Based on the analysis of above, we conclude that the input data with noised can be obtained by injecting the noise, such as Laplace noise or Gaussian noise, into the approximate and detail coefficients. Moreover, the noise injected into each input data is the sum of the noise injected into approximate and detail coefficients.

2.4.4. Injecting Noise via Haar Wavelet and Laplace Mechanism. In equation (12), we set $n_{k,i}$ as the noise with the Laplace distribution, as given in Definition 3. We have

$$n_{k,i} \sim Lap\left(\frac{\lambda}{2^k}\right), \tag{15}$$

where λ is the scale parameter of Laplace distribution and *k* denotes the *k*th decomposition level of lifting Haar wavelet transform.

According to equations (12) and (15), there is

$$n_m \sim Lap\left(\frac{\lambda}{2^l}\right) + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(Lap\left(\frac{\lambda}{2^k}\right) \cdot g_{k,i}\right), \tag{16}$$

where the symbols of n_m, $n_{k,i}$, $n_{l,0}$, and $g_{k,i}$ are the same as those in equation (12).

Using equations (12) and (16), we can describe the Laplace noise injected into Haar wavelet coefficients, as listed in Table 1.

Table 1: Laplace noise injected into Haar wavelet coefficients.

$n_0=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$+n_{1,1}$
$n_1=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$-n_{1,1}$
$n_2=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$+n_{1,2}$
$n_3=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$-n_{1,2}$
$n_4=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$+n_{1,3}$
$n_5=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$-n_{1,3}$
$n_6=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$+n_{1,4}$
$n_7=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$-n_{1,4}$
$n_{k,i}\sim$	Lap $(\lambda/2^3)$	Lap $(\lambda/2^3)$	Lap $(\lambda/2^2)$	Lap $(\lambda/2^1)$

According to Table 1, letting the range of the range query is n_0 to n_7, we have

$$\sum_{i=0}^{2^l-1} n_i = \sum_{i=0}^{7} n_i = 8\cdot n_{3,0} \sim Lap\left(8\cdot\frac{\lambda}{2^3}\right) = Lap(\lambda). \tag{17}$$

That means the sum of all noise injected into the input data is a noise with Laplace distribution with mean zero and scale λ.

Letting $\lambda = \Delta_1 f/\varepsilon$, then $\varepsilon = \Delta_1 f/\lambda$, according to Theorem 1, we conclude the $(\varepsilon, 0)$-differential privacy is preserved for the range query from n_0 to n_7 using Laplace mechanism.

According to Table 1, letting the range of the range query is n_1 to n_3, we have

$$n_1 + n_2 + n_3 = 3\cdot n_{3,0} + 3\cdot n_{3,1} - n_{2,1} - n_{1,1}. \tag{18}$$

Therefore,

$$\begin{aligned} n_1 + n_2 + n_3 &\sim Lap\left(3\cdot\frac{\lambda}{2^3}\right) + Lap\left(3\cdot\frac{\lambda}{2^3}\right) \\ &- Lap\left(\frac{\lambda}{2^2}\right) - Lap\left(\frac{\lambda}{2^1}\right). \end{aligned} \tag{19}$$

As we know, the sum of independent random Laplace variables is not a variable of Laplace distribution, so the compositive noise of range query of $n_1 + n_2 + n_3$ that injected into input data $x_1 + x_2 + x_3$ is not a noise with Laplace distribution. Therefore, we conclude that it is difficult to judge the level of differential privacy protection based on the Haar wavelet transform and Laplace mechanism.

To solve this problem, we consider adopting the Gaussian mechanism for the differential privacy via Haar wavelet transform in the next section.

3. Injecting Noise into Haar Wavelet Coefficients via Gaussian Mechanism

To inject Gaussian noise into Haar wavelet coefficients in Figure 3, we can set $n_{k,i}$ as the noise with the Gaussian distribution, as given in Definition 5.

Let

$$n_{k,i} \sim Gauss\left(\frac{3\sigma^2}{4^k}\right), \tag{20}$$

where $3\sigma^2/4^k$ is the variance of Gaussian distribution.

Table 2: Gaussian noise injected into wavelet coefficients.

$n_0=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$+n_{1,1}$
$n_1=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$-n_{1,1}$
$n_2=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$+n_{1,2}$
$n_3=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$-n_{1,2}$
$n_4=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$+n_{1,3}$
$n_5=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$-n_{1,3}$
$n_6=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$+n_{1,4}$
$n_7=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$-n_{1,4}$
$n_{k,i}\sim$	Gauss $(3\sigma^2/4^3)$	Gauss $(3\sigma^2/4^3)$	Gauss $(3\sigma^2/4^2)$	Gauss $(3\sigma^2/4^1)$

According to equations (12) and (20), we have

$$n_m \sim Gauss\left(\frac{3\sigma^2}{4^l}\right) + \sum_{n_{k,i}\in N\setminus\{n_{l,0}\}} \left(Gauss\left(\frac{3\sigma^2}{4^k}\right)\cdot g_{k,i}\right), \tag{21}$$

where the symbols of n_m, $n_{k,i}$, $n_{l,0}$, and $g_{k,i}$ are same as those in equation (12).

Using equations (20) and (21), we can describe the Gaussian noise injected into Haar wavelet coefficients, as listed in Table 2.

Theorem 3. *Suppose that X_1 and X_2 are independent random variables, and X_i has Gaussian distribution with mean zero and variance σ_i^2 for $i \in \{1, 2\}$. Then, $X_1 \pm X_2$ is Gaussian distribution with mean zero and variance $\sigma_1^2 + \sigma_2^2$; kX_1 is Gaussian distribution with mean zero and variance $(k\sigma_1)^2$.*

The proof of Theorem 3 will not be given because it is a basic property of Gaussian distribution.

According to Table 2 and Theorem 3, there is

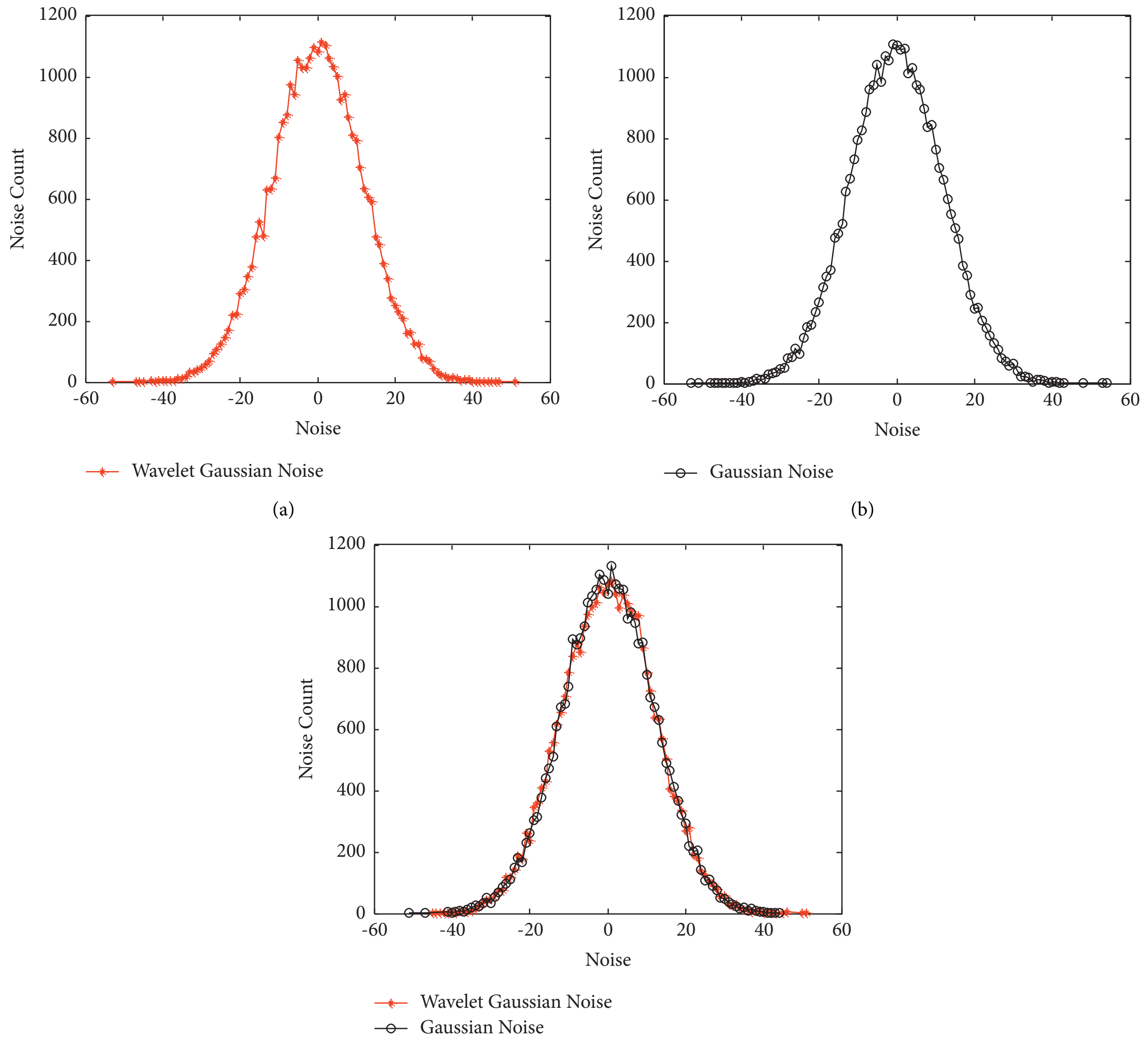
$$\sum_{i=0}^{2^l-1} n_i = \sum_{i=0}^{7} n_i = 8\cdot n_{3,0} = Gauss\left(8^2\cdot\frac{3\sigma^2}{4^3}\right) = Gauss\left(3\sigma^2\right). \tag{22}$$

That means the sum of all noise injected into the input data is a noise with Gaussian distribution. We analyze the distribution of the noise injected into each input data as follows.

Theorem 4. *Injecting Gaussian noise with variance $\sigma_k^2 = 3\sigma^2/4^k$ into the Haar wavelet coefficients in the kth decomposition level (the maximum decomposition level is l, as shown in Figure 3), the noise injected into each input data via Haar wavelet reconstruction is Gaussian noise with variance $(1 + 2/4^l)\,\sigma^2$.*

Proof. According to Definition 5, Theorem 3, Table 2, and equation (21), we have

$$\begin{aligned} n_m &\sim Gauss\left(\frac{3\sigma^2}{4^l}\right) \pm \sum_{i=l}^{1} Gauss\left(\frac{3\sigma^2}{4^i}\right), \\ &\sim Gauss\left(\left(\frac{1}{4^l} + \sum_{i=1}^{l}\left(1/4^i\right)\right)\cdot 3\sigma^2\right), \\ &\sim Gauss\left(\left(1 + \frac{2}{4^l}\right)\cdot\sigma^2\right). \end{aligned} \tag{23}$$



(a)

(b)

(c)

Figure 4: Comparison between injecting Gaussian noise into wavelet coefficients using Theorem 4 and injecting Gaussian noise into input data directly.

The proof is completed.

Using Theorems 2 and 4, we can obtain the (ε, δ)-differential privacy with variance $(1 + 2/4^l)\ \sigma^2$ for each input data under the framework of Gaussian mechanism via Haar wavelet transform. □

Theorem 5 (Differential privacy using Gaussian mechanism and Haar wavelet). *Let f be a function with ℓ_2-sensitivity, let $\varepsilon \in (0, 1)$ be arbitrary, and let $\sigma = \Delta_2 f \cdot \sqrt{2 \ln (1.25/\delta)}/\varepsilon$. The mechanism adopting Gaussian and Haar wavelet, which adds Gaussian noise with variance $\sigma_k^2 = 3\sigma^2/4^k$ into the Haar wavelet coefficients in the kth decomposition level (Figure 3), ensures (ε, δ)-differential privacy.*

Proof. In Theorem 4, if the Gaussian noise with variance $\sigma_k^2 = 3\sigma^2/4^k$is injected into the wavelet coefficients, the reconstructed data will be the one with the Gaussian noise with variance $\sqrt{1 + 2/4^l}\sigma$. According to Theorem 2, the condition of $\sqrt{1 + 2/4^l}\sigma \geq \Delta_2 f \cdot \sqrt{2 \ln (1.25/\delta)}/\varepsilon$ should be satisfied. Therefore, letting $\sigma = \Delta_2 f \cdot \sqrt{2 \ln (1.25/\delta)}/\varepsilon$, the condition is met and the proof is completed.

We simulate the process of injecting Gaussian noise with = 12 into wavelet coefficients with 15-level decomposition using Theorem 4 and injecting Gaussian noise into input data directly and draw the noise-count figures as follows.

Figure 4(a) shows the count of the noise injected into input data by injecting the Gaussian noise with $\sigma = 12$ into Haar wavelet coefficients with 15-level decomposition using equation (20); Figure 4(b) denotes the noise count by injecting Gaussian noise with mean zero and variance $(1 + 2/4^l)\ \sigma^2$ into the input data directly. In Figure 4(c), we draw the two curves together and we find that they are almost overlapped. Figure 4(c) shows that the method injecting Gaussian noise into Haar wavelet coefficients has the same level of differential privacy protection as the method injecting Gaussian noise into the input data directly. In this study, we will focus on the application of range query.

According to Theorems 3–5, we find that the distribution of noise for the range query using Gaussian mechanism and Haar wavelet is a Gaussian distribution. Therefore, we can calculate the variance of noise easily, for example, as listed in Table 2, we have

$$\begin{aligned} n_1 + n_2 + n_3 &= n_{3,0} + n_{3,1} + n_{2,1} - n_{1,1} \\ &\quad + n_{3,0} + n_{3,1} - n_{2,1} + n_{1,2} \\ &\quad + n_{3,0} + n_{3,1} - n_{2,1} - n_{1,2} \\ &= 3n_{3,0} + 3n_{3,1} - n_{2,1} - n_{1,1}, \\ &\sim 3Gauss\left(\frac{3\sigma^2}{4^3}\right) + 3Gauss\left(\frac{4\sigma^2}{4^3}\right) \\ &\quad - Gauss\left(\frac{3\sigma^2}{4^2}\right) - Gauss\left(\frac{3\sigma^2}{4^1}\right), \\ &\sim Gauss\left(3^2 \cdot \frac{3\sigma^2}{4^3} + 3^2 \cdot \frac{3\sigma^2}{4^3} + \frac{3\sigma^2}{4^2} + \frac{3\sigma^2}{4^1}\right), \\ &\sim Gauss\left(\frac{57}{32}\sigma^2\right). \end{aligned} \tag{24}$$

From this example, we find that some noises (such as $n_{2,1}$ and $-n_{2,1}$, $n_{1,2}$ and $-n_{1,2}$) are eliminated by the operation of addition. According to Theorem 4, the variance of noise injected into each input data should be $(1 + 2/4^3)\sigma^2$ for $l = 3$. The total noise variance is $3 * (1 + 2/4^3)\sigma^2 = (99/32) * \sigma^2$. Compared with equation (24), we conclude that, for the range query, the noise actually added into a certain range of the original data is much less than the sum of the noise at each data. Therefore, it is a very important property for Gaussian mechanism to be used on range query. □

4. Noise of Range Query under the Framework of Haar Wavelet and Gaussian Mechanism

In this section, we discuss how to compute the noise of the range query under the framework of Haar wavelet transform and Gaussian mechanism. First, we give the computing method for the variance of range query in detail. Second, the interval of the range query when obtaining the maximum variance is introduced. Third, to speed up the computing of maximum variance, we observe the results of the range-count interval when getting the maximum variance and give a speed computing method. Finally, we give a coarse estimation of the maximum variance of range query as a function expression.

4.1. Computing Method for the Variance of Range Query. In Figure 3, we choose the Gaussian noise and inject them into the approximate coefficient and each wavelet coefficient.

Table 3: Noise variance and decomposition level k.

k	Variance of noise	Number of wavelet coefficients
3	$3\sigma^2/4^3$	2^{3-3}
2	$3\sigma^2/4^2$	2^{3-2}
1	$3\sigma^2/4^1$	2^{3-1}

The variance of Gaussian noise injected into approximate coefficient is $3\sigma^2/4^3$. The variance of Gaussian noise injected into each wavelet coefficient is $3\sigma^2/4^k$ for level $k\,(k \in [1, 3])$. The relationship between decomposition level and variance of noise is listed in Table 3.

The noise-sum of range query via Haar wavelet transform for interval S can be given by the following equation (Figure 3):

$$n_{sum} = |S| \cdot n_{l,0} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(n_{k,i} \cdot \left(\alpha\left(n_{k,i}\right) - \beta\left(n_{k,i}\right)\right)\right), \tag{25}$$

where S is the interval of any range query, $n_{k,i}$ presents the ith noise injected into wavelet coefficient in kth decomposition level, $\alpha(n_{k,i})$ denotes the number of left leaves in the left subtree of $n_{k,i}$ that are contained in S, and $\beta(n_{k,i})$ denotes the number of right leaves in the right subtree of $n_{k,i}$ that are contained in S (Figure 3).

Now we analyze the noise variance of range query. According to equation (20), we know that the noise injected into approximate coefficient is $n_{l,0}$ and its variance is $3\sigma^2/4^l$; the noise injected into wavelet coefficient is $n_{k,i}$ and its variance is $3\sigma^2/4^k$. Therefore, according to Theorem 3, we can compute the noise-variance of range query by replacing $n_{l,0}$ and $n_{k,i}$ with $3\sigma^2/4^l$ and $3\sigma^2/4^k$ in equation (25), respectively.

$$\begin{aligned} \sigma_{sum}^2 &= |S|^2 \cdot \frac{3\sigma^2}{4^l} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(\frac{3\sigma^2}{4^k}\left(\alpha\left(n_{k,i}\right) - \beta\left(n_{k,i}\right)\right)^2\right) \\ &= \left(|S|^2 \cdot \frac{1}{4^l} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(\frac{1}{4^k}\left(\alpha\left(n_{k,i}\right) - \beta\left(n_{k,i}\right)\right)^2\right)\right) \cdot 3\sigma^2. \end{aligned} \tag{26}$$

To compute the value of σ_{sum}^2, we need to calculate the values of $\alpha(n_{k,i})$ and $\beta(n_{k,i})$ firstly. In Figure 3, the length of interval of leaves in the subtree of $n_{k,i}$ is 2^k. The left point of this interval has the subscript $(i-1) \cdot 2^k$ and the right point of this interval has the subscript $i \cdot 2^k - 1$. Therefore, the subtree of $n_{k,i}$ has the subscript interval of leaves.

$$S_{n_{k,i}} = \left[(i-1) \cdot 2^k, i \cdot 2^k - 1\right]. \tag{27}$$

For example, the wavelet coefficient $n_{2,2}$ in Figure 3 has the subscript interval of leaves [4, 7].

According to equation (27), we can obtain the left-half interval $[\alpha_L, \alpha_R]$and right-half interval $[\beta_L, \beta_R]$ of $S\_n_{k,i}$:

$$\begin{aligned} \left[\alpha_L, \alpha_R\right] &= \left[(i-1) \cdot 2^k, i \cdot 2^k - 2^{k-1} - 1\right], \\ \left[\beta_L, \beta_R\right] &= \left[i \cdot 2^k - 2^{k-1}, i \cdot 2^k - 1\right]. \end{aligned} \tag{28}$$

```
     Input: the maximum decomposition level $l$
     Output: $\sigma^{2}_{\text{sumMax}}$, $S_{\text{L}}$ and $S_{\text{R}}$ (for $\sigma^{2}_{\text{sumMax}}$)
(1)  $\sigma^{2}_{\text{sumMax}} = 0$
(2)  For $S_{\text{L}} = 0$ to $2^{l}-1$
(3)    For $S_{\text{R}} = S_{\text{L}}$ to $2^{l}-1$
(4)      $sum = 0$
(5)      For $k = 1$ to $l$
(6)        For $i = 1$ to $2^{l-\text{k}}$
(7)          Compute $\alpha_L$, $\alpha_R$, $\beta_L$, $\beta_R$ of $n_{k,\ i}$ using equations (38) and (39)
(8)          If $S_R < \alpha_L$ or $S_L > \alpha_R$ then $\alpha\ (n_{k,\ i}) = 0$
(9)          Elseif $S_L \geq \alpha_L$ and $S_R \leq \alpha_R$ then $\alpha\ (n_{k,\ i}) = S_R - S_L + 1$
(10)         Elseif $S_L < \alpha_L$ and $S_R > \alpha_R$ then $\alpha\ (n_{k,\ i}) = \alpha_R - \alpha_L + 1$
(11)         Elseif $S_L < \alpha_L$ and $\alpha_L \leq S_R \leq \alpha_R$ then $\alpha\ (n_{k,\ i}) = S_R - \alpha_L + 1$
(12)         Elseif $S_R > \alpha_R$ and $\alpha_L \leq S_L \leq \alpha_R$ then $\alpha\ (n_{k,\ i}) = \alpha_R - S_L + 1$
(13)         End If
(14)         If $S_R < \beta_L$ or $S_{\text{L}} > \beta_R$ then $\beta\ (n_{k,\ i}) = 0$
(15)         Elseif $S_{\text{L}} \geq \beta_L$ and $S_R \leq \beta_R$ then $\beta\ (n_{k,\ i}) = S_R - S_L + 1$
(16)         Elseif $S_{\text{L}} < \beta_L$ and $S_R > \beta_R$ then $\beta\ (n_{k,\ i}) = \beta_R - \beta_L + 1$
(17)         Elseif $S_{\text{L}} < \beta_L$ and $\beta_L \leq S_R \leq \beta_{\text{R}}$ then $\beta\ (n_{k,\ i}) = S_R - \beta_L + 1$
(18)         Elseif $S_R > \beta_{\text{R}}$ and $\beta_L \leq S_L \leq \beta_{\text{R}}$ then $\beta\ (n_{k,\ i}) = \beta_R - S_L + 1$
(19)         End If
(20)         Compute $sum = sum + (1/4^{k})\ (\alpha\ (n_{k,\ i}) - \beta\ (n_{k,\ i}))^{2}$
(21)       End For
(22)     End For
(23)     Compute $\sigma^{2}_{\text{sum}}$ using $sum$ and (26)
(24)     If $\sigma^{2}_{\text{sum}} > \sigma^{2}_{\text{sumMax}}$
(25)       $\sigma^{2}_{\text{sumMax}} = \sigma^{2}_{\text{sum}}$
(26)     End IF
(27)   End For
(28) End For
(29) Import $\sigma^{2}_{\text{sumMax}}$, $S_L$ and $S_R$.
```

Algorithm 1: Compute $\sigma^{2}_{\text{sumMax}}$for fix l ($l \geq 2$).

Therefore, $\alpha(n_{k,i})$ and $\beta(n_{k,i})$ can be given by computing the number of intersection between S and $[\alpha_L, \alpha_R]$, S and $[\beta_L, \beta_R]$, respectively.

$$\begin{aligned}\alpha(n_{k,i}) &= |S \cap [\alpha_L, \alpha_R]|,\\ \beta(n_{k,i}) &= |S \cap [\beta_L, \beta_R]|.\end{aligned} \tag{29}$$

Let $S = [S_L, S_R]$, where S_L and S_R denote the left and right points of the given range query interval, respectively. Therefore, we have

$$\alpha(n_{k,i}) = |[S_L, S_R] \cap [\alpha_L, \alpha_R]|, \tag{30}$$

$$\beta(n_{k,i}) = |[S_L, S_R] \cap [\beta_L, \beta_R]|, \tag{31}$$

where $k \in [1, l]$ and $i \in [1, 2^{l-k}]$ (Figure 3).

4.2. Maximum Variance of Range Query. The aim of this study is to obtain the maximum value of range query for any fixed maximum decomposition level l (the number of input data is 2^l). According to equation (26), we have

$$\sigma^2_{sumMax} = \max\left(|S|^2 \cdot \frac{1}{4^l} + \sum_{n_{k,i} \in N \setminus \{n_{l,0}\}} \left(\frac{1}{4^k}\left(\alpha(n_{k,i}) - \beta(n_{k,i})\right)^2\right)\right) \cdot 3\sigma^2. \tag{32}$$

In equation (32), the given parameter is l. To compute the value of σ^2_{sumMax}, we need to calculate any range query interval $S = [S_L, S_R]$ in all data and obtain the count $\alpha(c_{k,i})$ and $\beta(c_{k,i})$ using equations (30) and (31). We give the pseudocode of computing σ^2_{sumMax} as follows:

Algorithm 1 illustrates the details of the algorithm of computing $\sigma^{2}_{\text{sumMax}}$ for fix l ($l \geq 2$). Step 1 is the initialization of $\sigma^{2}_{\text{sumMax}}$. Steps 2 to 3 are the range loop of S_L and S_R. Step 5 is the loop of the subscript of decomposition level k. Step 6 is the loop of the subscript of the wavelet coefficient in kth decomposition level. Step 7 is the computation of the α_L, α_R, β_L, and β_R of $n_{k,\ i}$. Steps 8 to 13 denote the computation of α ($n_{k,\ i}$). Steps 14 to 19 denote the computation of β ($n_{k,\ i}$). Step 20 is the computation of right part of σ^{2}_{sum} using equation (26). Step 23 is the computation of σ^{2}_{sum} using equation (26). Steps 24 to 26 denote the computation of $\sigma^{2}_{\text{sumMax}}$ using equation (34). Step 29 denotes the output of Algorithm 1.

According to Algorithm 1, we calculate the values of $\sigma^{2}_{\text{sumMax}}$, S_L, and S_R, as listed in Table 4.

In Table 4, $S_R - S_L + 1$ denotes the length of interval for the $\sigma^{2}_{\text{sumMax}}$. It will take a very long time to compute the $\sigma^{2}_{\text{sumMax}}$ using Algorithm 1 when $l > 14$, so we need to find some method to speed up Algorithm 1.

4.3. Speeding Algorithm for Computing the Maximum Variance of Range Query. Observing the values of S_L and S_R in

Table 4: Max-variance of range query via Haar wavelet and Gaussian mechanism (l from 2 to 14).

l	$\sigma^{2}_{\text{sumMax}}$ $(*\sigma^2)$	S_L	S_R	$S_R - S_L + 1$
2	3.000000	0	3	4
3	3.562500	1	6	6
4	4.265625	1	14	14
5	4.910156	3	28	26
6	5.586914	5	58	54
7	6.248291	11	116	106
8	6.917542	21	234	214
9	7.582901	43	468	426
10	8.250217	85	938	854
11	8.916558	171	1876	1706
12	9.583388	341	3754	3414
13	10.249973	683	7508	6826
14	10.916680	1365	15018	13654

Table 5: Statistic results of S_L and S_R.

l	S_L		S_R	
2	0		3	
3	1	$\mathbf{0} \times 2 + 1$	6	$\mathbf{3} \times 2$
4	1	$\mathbf{1} \times 2 - 1$	14	$\mathbf{6} \times 2 + 2$
5	3	$\mathbf{1} \times 2 + 1$	28	$\mathbf{14} \times 2$
6	5	$\mathbf{3} \times 2 - 1$	58	$\mathbf{28} \times 2 + 2$
7	11	$\mathbf{5} \times 2 + 1$	116	$\mathbf{58} \times 2$
8	21	$\mathbf{11} \times 2 - 1$	234	$\mathbf{116} \times 2 + 2$
9	43	$\mathbf{21} \times 2 + 1$	468	$\mathbf{234} \times 2$
10	85	$\mathbf{43} \times 2 - 1$	938	$\mathbf{468} \times 2 + 2$
11	171	$\mathbf{85} \times 2 + 1$	1876	$\mathbf{938} \times 2$
12	341	$\mathbf{171} \times 2 - 1$	3754	$\mathbf{1876} \times 2 + 2$
13	683	$\mathbf{341} \times 2 + 1$	7508	$\mathbf{3754} \times 2$
14	1365	$\mathbf{683} \times 2 - 1$	15018	$\mathbf{7508} \times 2 + 2$

Table 4, we give some statistical rules to compute them directly in Table 5.

According to Table 5, we can compute $\sigma^{2}_{\text{sumMax}}$ using the following algorithm:

Algorithm 2 illustrates the details of the algorithm of computing $\sigma^{2}_{\text{sumMax},l}$ for any l $(l \geq 2)$. Step 1 is the initialization of S_L and S_R. Step 2 is the loop of the maximum decomposition level l. Steps 3 to 7 denote the computation of S_L and S_R according to the maximum decomposition level l. Step 10 is the loop of the subscript of decomposition level k. Step 11 is the loop of the subscript of the wavelet coefficient in kth decomposition level. Step 12 is the computation of the α_L, α_R, β_L, and β_R of $n_{k,\ i}$. Steps 13 to 18 denote the computation of α $(n_{k,\ i})$. Steps 19 to 24 denote the computation of β $(n_{k,\ i})$. Step 25 is the computation of right part of σ^{2}_{sum} using equation (26). Step 28 is the computation of σ^{2}_{sum} using equation (26). Step 30 denotes the output of Algorithm 2 for any l.

S_L and S_R can also be given directly by simplifying the results in Table 5.

If l is an even number,

$$S_L = \frac{2^{l-2} - 1}{3}, S_R = \frac{11 \times 2^{l-2} - 2}{3}. \tag{33}$$

If l is an odd number,

$$S_L = \frac{2^{l-2} + 1}{3}, S_R = \frac{11 \times 2^{l-2} - 4}{3}. \tag{34}$$

Therefore, we give the values of $\sigma^{2}_{\text{sumMax}}$, S_L, S_R, and $S_R - S_L + 1$ for l from 2 to 30 in Table 6.

In Table 6, S_L and S_R denote the left and right points of the range query interval when the $\sigma^{2}_{\text{sumMax}}$ is met. $S_R - S_L + 1$ denotes the length of interval for the $\sigma^{2}_{\text{sumMax}}$. In Table 6, we observe that $\sigma^{2}_{\text{sumMax}}$ will increase about (2/3) σ^2 if the parameter l increases 1.

4.4. Coarse Estimation of the Maximum Variance of Range Query

4.4. Coarse Estimation of the Maximum Variance of Range Query. In previous sections, the maximum variance of range queries via Gaussian mechanism and Haar wavelet transform is given for any l. But it is obtained using a computer program, not from a function expression. In this section, the coarse estimation of the maximum variance is given in Theorem 6, and it is a function expression with parameters l and σ^2.

Theorem 6 (Coarse estimation of the maximum variance). *Let N be a set of independent Gaussian noise $n_{k,i} \in N$ with a variance $3\sigma^2/4^k$, which is injected into one-dimensional Haar wavelet coefficients and approximate coefficient (Figure 3). Suppose $l = \log_2|N|$, that means the number of Gaussian noise injected into Haar wavelet*

```
Input:
Output: $l$, $\sigma^2_{\text{sumMax},\ l}$, $S_L$ and $S_R$ (for $l$ and $\sigma^2_{\text{sumMax},\ l}$)
$S_L = 0$, $S_R = 3$ (for $l = 2$)
For $l = 3$ to 30
    If $l$ is odd then
        $\mathbf{S_L = S_L \times 2 + 1,\ S_R = S_R \times 2}$
    Elseif $l$ is even then
        $\mathbf{S_L = S_L \times 2 - 1,\ S_R = S_R \times 2 + 2}$
    End If
    $\sigma^2_{\text{sumMax},\ l} = 0$
    $sum = 0$
    For $k = 1$ to $l$
        For $i = 1$ to $2^{l-k}$
            Compute $\alpha_L$, $\alpha_R$, $\beta_L$, $\beta_R$ of $n_{k,\ i}$ using equations (38) and (39)
            If $S_R < \alpha_L$ or $S_L > \alpha_R$ then $\alpha\ (n_{k,\ i}) = 0$
            Elseif $S_L \ge \alpha_L$ and $S_R \le \alpha_R$ then $\alpha\ (n_{k,\ i}) = S_R - S_L + 1$
            Elseif $S_L < \alpha_L$ and $S_R > \alpha_R$ then $\alpha\ (n_{k,\ i}) = \alpha_R - \alpha_L + 1$
            Elseif $S_L < \alpha_L$ and $\alpha_L \le S_R \le \alpha_R$ then $\alpha\ (n_{k,\ i}) = S_R - \alpha_L + 1$
            Elseif $S_R > \alpha_R$ and $\alpha_L \le S_L \le \alpha_R$ then $\alpha\ (n_{k,\ i}) = \alpha_R - S_L + 1$
            End If
            If $S_R < \beta_L$ or $S_L > \beta_R$ then $\beta\ (n_{k,\ i}) = 0$
            Elseif $S_L \ge \beta_L$ and $S_R \le \beta_R$ then $\beta\ (n_{k,\ i}) = S_R - S_L + 1$
            Elseif $S_L < \beta_L$ and $S_R > \beta_R$ then $\beta\ (n_{k,\ i}) = \beta_R - \beta_L + 1$
            Elseif $S_L < \beta_L$ and $\beta_L \le S_R \le \beta_R$ then $\beta\ (n_{k,\ i}) = S_R - \beta_L + 1$
            Elseif $S_R > \beta_R$ and $\beta_L \le S_L \le \beta_R$ then $\beta\ (n_{k,\ i}) = \beta_R - S_L + 1$
            End If
            Compute $sum = sum + (1/4^k)\ (\alpha\ (n_{k,\ i}) - \beta\ (n_{k,\ i}))^2$
        End For
    End For
    Compute $\sigma^2_{\text{sum}}$ using $sum$ and (26)
    $\sigma^2_{\text{sumMax},\ l} = \sigma^2_{\text{sum}}$
    Import $l$, $\sigma^2_{\text{sumMax},\ l}$, $S_L$ and $S_R$
End For.
```

Algorithm 2: Compute $\sigma^2_{\text{sumMax},\ l}$ for any l ($l \ge 2$).

coefficients and approximate coefficient is 2^l *(the number of input data is also* 2^l*). Let M be the noisy data reconstructed from* $C + N$ *(C is the set of one-dimensional Haar wavelet coefficients of the input data, refer to Figure 2). Then, for any range query answered using M, the variance of noise in the answer is at most* $((6l + 9)/4)\sigma^2$.

Proof. Referring to Figure 3 and equation (26), we observe that for any noise $n_{k,i}$, if none of the leaves under $n_{k,i}$ is contained in S, then there is $\alpha(n_{k,i}) = \beta(n_{k,i}) = 0$. On the other hand, if all leaves under $n_{k,i}$ are covered by S, then $\alpha(n_{k,i}) = \beta(n_{k,i}) = 2^{k-1}$. Therefore, $\alpha(n_{k,i}) - \beta(n_{k,i}) \ne 0$, if and only if the left or right subtree of $n_{k,i}$ partially intersects S. At any level of the decomposition tree except for the lth level, there exist at most two such noises. At the level l, at most one such noise that letting the condition $\alpha(n_{k,i}) - \beta(n_{k,i}) \ne 0$ be sufficient.

Considering a noise $n_{k,i}$ at level k ($k \in [1, l]$), such that $\alpha(n_{k,i}) - \beta(n_{k,i}) \ne 0$. Since the left (right) subtree of $n_{k,i}$ contains at most 2^{k-1} leaves, we have $\alpha(n_{k,i}), \beta(n_{k,i}) \in [0, 2^{k-1}]$. So, there is $|\alpha(n_{k,i}) - \beta(n_{k,i})| \le 2^{k-1}$. Therefore, the variance of the range query about the noise $n_{k,i}$ ($k \in [1, l]$) at most is

$$\left(\alpha\left(n_{k,i}\right) - \beta\left(n_{k,i}\right)\right)^2 \cdot \left(\frac{3\sigma^2}{4^k}\right) \le 4^{k-1} \cdot \left(\frac{3\sigma^2}{4^k}\right) = 3\sigma^2/4. \tag{35}$$

On the other hand, the noise in the approximate coefficient ($n_{l,0}$) has a variance at most:

$$\left(2^l\right)^2 \cdot \left(\frac{3\sigma^2}{4^l}\right) = 4^l \cdot \left(\frac{3\sigma^2}{4^l}\right) = 3\sigma^2. \tag{36}$$

Therefore, the total variance injected into wavelet coefficients of 1 to $l - 1$ level is $2 \cdot (l - 1) \cdot 3\sigma^2/4$, and the variance injected into wavelet coefficients of level l is $1 \cdot 3\sigma^2/4$. According to equation (26), the variance of noise at most is

$$3\sigma^2 + 2\bullet(l - 1) \cdot \frac{3\sigma^2}{4} + 1\bullet\frac{3\sigma^2}{4} = \left(\frac{6l + 9}{4}\right)\sigma^2, \tag{37}$$

which completes the proof.

This conclusion in Theorem 6 can also be obtained by observing Table 2. Now, we give the intuitive explanation of Theorem 6.

According to Table 2, we can give a coarse estimation of the maximum variance of range query. In Table 2, we insert a row at the bottom to calculate the maximum variance sum for each column. The new table is shown as follows.

Table 6: Max-variance of range query via Haar wavelet and Gaussian mechanism (any l).

l	$\sigma\ ^2{}_{\text{sumMax}}\ (*\sigma^2)$	$S\ _L$	$S\ _R$	$S\ _R - S_L + 1$
2	3.000000	0	3	4
3	3.562500	1	6	6
4	4.265625	1	14	14
5	4.910156	3	28	26
6	5.586914	5	58	54
7	6.248291	11	116	106
8	6.917542	21	234	214
9	7.582901	43	468	426
10	8.250217	85	938	854
11	8.916558	171	1876	1706
12	9.583388	341	3754	3414
13	10.249973	683	7508	6826
14	10.916680	1365	15018	13654
15	11.583327	2731	30036	27306
16	12.250003	5461	60074	54614
17	12.916665	10923	120148	109226
18	13.583334	21845	240298	218454
19	14.250000	43691	480596	436906
20	14.916667	87381	961194	873814
21	15.583333	174763	1922388	1747626
22	16.250000	349525	3844778	3495254
23	16.916667	699051	7689556	6990506
24	17.583333	1398101	15379114	13981014
25	18.250000	2796203	30758228	27962026
26	18.916667	5592405	61516458	55924054
27	19.583333	11184811	123032916	111848106
28	20.250000	22369621	246065834	223696214
29	20.916667	44739243	492131668	447392426
30	21.583333	89478485	984263338	894784854

In Table 7, each value of the last row is given by calculating the maximum sum of variance of some continued parts for each column according to Theorem 3. For example, the value of column 2 in the last row, $3\sigma^2$, is calculated by range from n_0 to n_7:

$$8n_{3,0} \sim 8Gauss\left(\frac{3\sigma^2}{4^3}\right) = Gauss\left(8^2 \cdot \frac{3\sigma^2}{4^3}\right) = Gauss\left(3\sigma^2\right). \tag{38}$$

The value of column 3 in the last row, $3\sigma^2/4$, is calculated by range from n_0 to n_3 or from n_4 to n_7:

$$\pm 4n_{3,1} \sim 4Gauss\left(\frac{3\sigma^2}{4^3}\right) = Gauss\left(4^2 \cdot \frac{3\sigma^2}{4^3}\right) = Gauss\left(\frac{3\sigma^2}{4}\right). \tag{39}$$

The value of column 4 in the last row, $3\sigma^2/2$, is calculated by range from n_2 to n_5:

$$\begin{aligned} -2n_{2,1} + 2n_{2,2} &\sim 2Gauss\left(\frac{3\sigma^2}{4^2}\right) + 2Gauss\left(\frac{3\sigma^2}{4^2}\right), \\ &\sim Gauss\left(2^2 \cdot \frac{3\sigma^2}{4^2}\right) + Gauss\left(2^2 \cdot \frac{3\sigma^2}{4^2}\right), \\ &\sim Gauss\left(2 \cdot 2^2 \cdot \frac{3\sigma^2}{4^2}\right), \\ &\sim Gauss\left(\frac{3\sigma^2}{2}\right). \end{aligned} \tag{40}$$

The value of column 5 in the last row, $3\sigma^2/2$, is calculated by range from n_1 to n_2 or from n_3 to n_4 or from n_5 to n_6:

$$\begin{aligned} -n_{1,1} + n_{1,2} &\sim Gauss\left(3\sigma^2/4 + 3\sigma^2/4\right) = Gauss\left(3\sigma^2/2\right) \\ -n_{1,2} + n_{1,3} &\sim Gauss\left(3\sigma^2/4 + 3\sigma^2/4\right) = Gauss\left(3\sigma^2/2\right). \\ -n_{1,3} + n_{1,4} &\sim Gauss\left(3\sigma^2/4 + 3\sigma^2/4\right) = Gauss\left(3\sigma^2/2\right) \end{aligned} \tag{41}$$

Therefore, we can obtain an estimation of the maximum variance of range query as follows:

$$\sigma^2_{\text{sumMaxEstim}} = 3\sigma^2 + \frac{3}{4}\sigma^2 + \frac{3}{2}\sigma^2 + \frac{3}{2}\sigma^2 = \frac{27}{4}\sigma^2 = \frac{6*3+9}{4}\sigma^2. \tag{42}$$

In Table 7, the wavelet decomposition level is 3 (the number of input data is 2^3). Supposing that the decomposition level is l, then we can give an estimation of the maximum variance of range query:

$$\sigma^2_{\text{sumMaxEstim}} = \left(\frac{6l+9}{4}\right)\sigma^2. \tag{43}$$

Note that equation (43) gives the same result with the conclusion in Theorem 6.

Comparing the estimation maximum variances (equation (43)) with the real maximum variances (Table 6), we find that

$$\sigma^2_{\text{sumMax}} \ll \sigma^2_{\text{sumMaxEstim}}. \tag{44}$$

Table 7: Coarse estimation of max-variance for range query.

$n_{0}=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$+n_{1,1}$
$n_{1}=$	$n_{3,0}$	$+n_{3,1}$	$+n_{2,1}$	$-n_{1,1}$
$n_{2}=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$+n_{1,2}$
$n_{3}=$	$n_{3,0}$	$+n_{3,1}$	$-n_{2,1}$	$-n_{1,2}$
$n_{4}=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$+n_{1,3}$
$n_{5}=$	$n_{3,0}$	$-n_{3,1}$	$+n_{2,2}$	$-n_{1,3}$
$n_{6}=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$+n_{1,4}$
$n_{7}=$	$n_{3,0}$	$-n_{3,1}$	$-n_{2,2}$	$-n_{1,4}$
$n_{k,i}\sim$	Gauss $(3\sigma^2/4^3)$	Gauss $(3\sigma^2/4^3)$	Gauss $(3\sigma^2/4^2)$	Gauss $(3\sigma^2/4^1)$
$\sigma^{2}_{\text{sumMax}}=$	$3\sigma^2$	$+3\sigma^2/4$	$+3\sigma^2/2$	$+3\sigma^2/2$

$\sigma^2_{sumMaxEstim}$ provides a function expression using parameters l and σ^2 for the maximum variance of range query via Haar wavelet transform, but it has a very large error comparing the real maximum variance, comparing equation (43) with Table 6. Therefore, for the analysis of the practical applications, we prefer to use the maximum variance σ^2_{sumMax}, as listed in Table 6. □

5. Experimental Verification

This section introduces the experimental verification of the proposed framework, that is, the computing of maximum variance for range query based on Gaussian mechanism and lifting Haar wavelet. We use the dataset age, which contains census records of individuals from the United States. The age has 107, 974, and 787 records, each of which corresponds to the age of an individual, extracted from the IPUM's census data of the United States [40]. The ages range from 0 to 135 and just have 128 values (ages 121, 123, 127, 128, 131, 132, 133, and 134 are empty). We count the number of each age and give the histogram of age as the input file of our experiments. Given a query length L, we test all the possible range queries with length L and report the maximum variance of the range query for input data.

The noise injected into the input data via Gaussian mechanism is Gaussian noise. The variance of Gaussian noise is the sample variance. Therefore, the sample variance is adopted in this study and is computed by the following equation:

$$Var(n) = \frac{1}{m-1}\sum_{i=1}^{m}\left(n_i - \frac{1}{m}\sum_{j=1}^{m} n_j\right)^2, \tag{45}$$

where n_i (or n_j) is the noise injected into the input data and m is the number of the input data.

We research the maximum variance of the range query of noise when ε chosen in the set {0.5, 0.75, 1.0, 1.25} and δ chosen in the set {0.1, 0.01, 0.001}. For each special ε, we draw the maximum variance of the range count of noise using Gaussian mechanism and Gaussian mechanism with Haar wavelet transform when δ is equal to 0.1, 0.01, and 0.001.

For any ε and δ, we can calculate the σ of Gaussian noise using the equation $\sigma = \sqrt{2\ln(1.25/\delta)}/\varepsilon$ in Theorem 5, and the results are given in Table 8.

To compute the variance, we inject the Gaussian noise into the input data or the Haar wavelet coefficients 10000 times. To compute the maximum variance of the range query with fixed ε and δ, each variance of the range query for range size k needs to be computed firstly. Then, the maximum value of variance for range size k can be given by comparing all the variance of the k-range queries.

Table 8: σ for some ε and δ.

ε	δ	σ
0.5	0.1	4.4951
0.5	0.01	6.2150
0.5	0.001	7.5530
0.75	0.1	2.9967
0.75	0.01	4.1433
0.75	0.001	5.0353
1.0	0.1	2.2475
1.0	0.01	3.1075
1.0	0.001	3.7765
1.25	0.1	1.7980
1.25	0.01	2.4860
1.25	0.001	3.0212

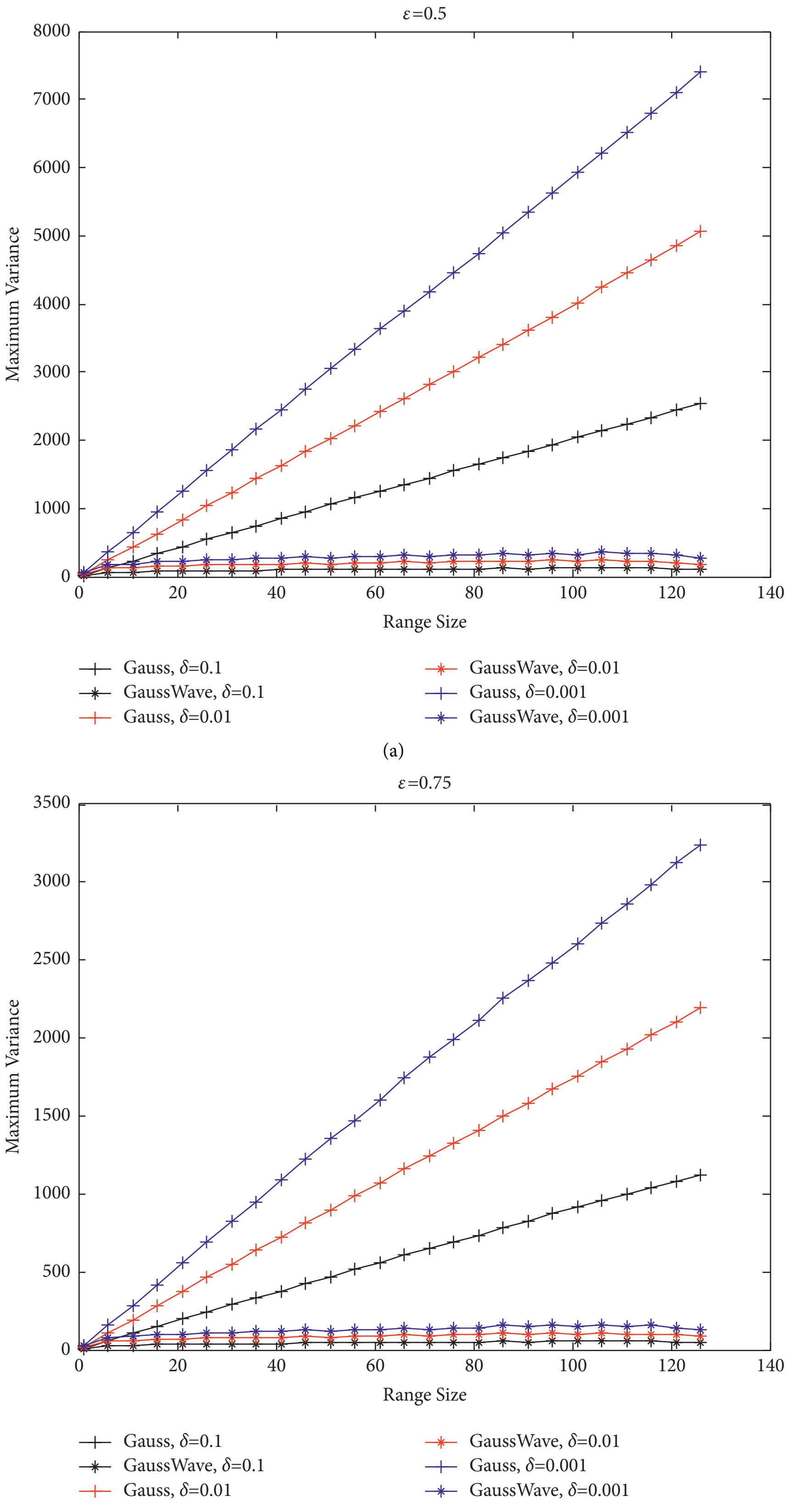
For input data with length 128 (such as 128 histogram), we can draw the maximum variance diagram of the range query using Gaussian mechanism and Gaussian mechanism with Haar wavelet transform for any range sizes, as shown in Figure 5.

In Figure 5, "Gauss" means injecting noise into each histogram data via Gaussian mechanism directly and then gets the noise of range query by the operation of addition. First, "GaussWave" denotes injecting noise into the lifting Haar wavelet coefficients using Theorem 5. Second, the noise injected into each histogram is obtained by the inverse wavelet transform. Finally, the range query for any range size is obtained by injecting the noise together.

In Figure 5, we observe that the maximum variance is increasing linearly with the "range size" for "Gauss." In Figure 5, for any ε and δ, the maximum variance of the noise using "GaussWave" method is far less than the noise using "Gauss" method.

To observe the variation tendency of "GaussWave" in Figure 5, we just draw the maximum variance diagram of the range query using Gaussian mechanism with Haar wavelet transform, as shown in Figure 6.

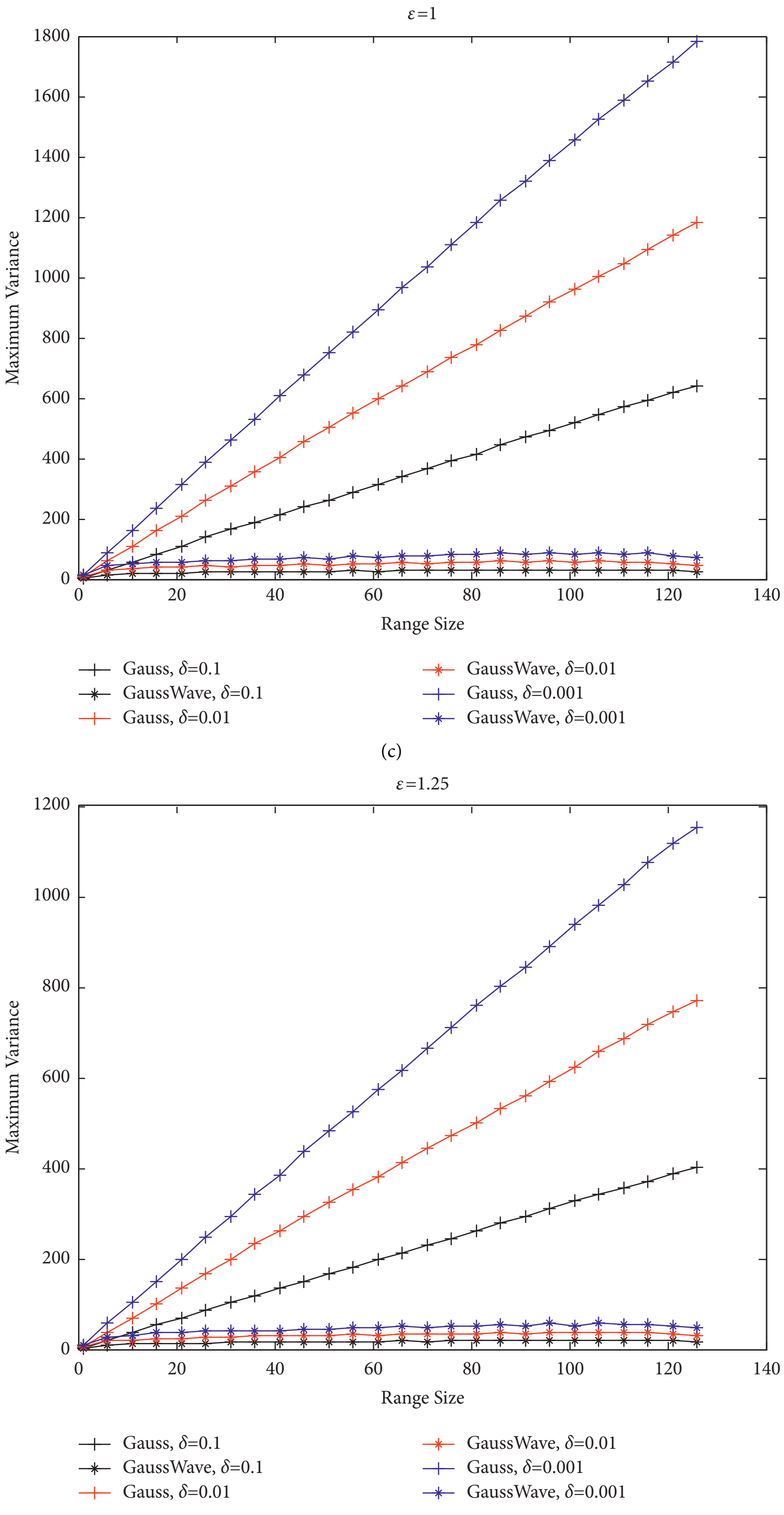
In Figure 6, for any ε and δ, the maximum variance of the noise using "GaussWave" method increases with the increase of range size before it gets the maximum value, and it will decrease with the increase of range size after it has gotten the maximum value.



(a)

(b)

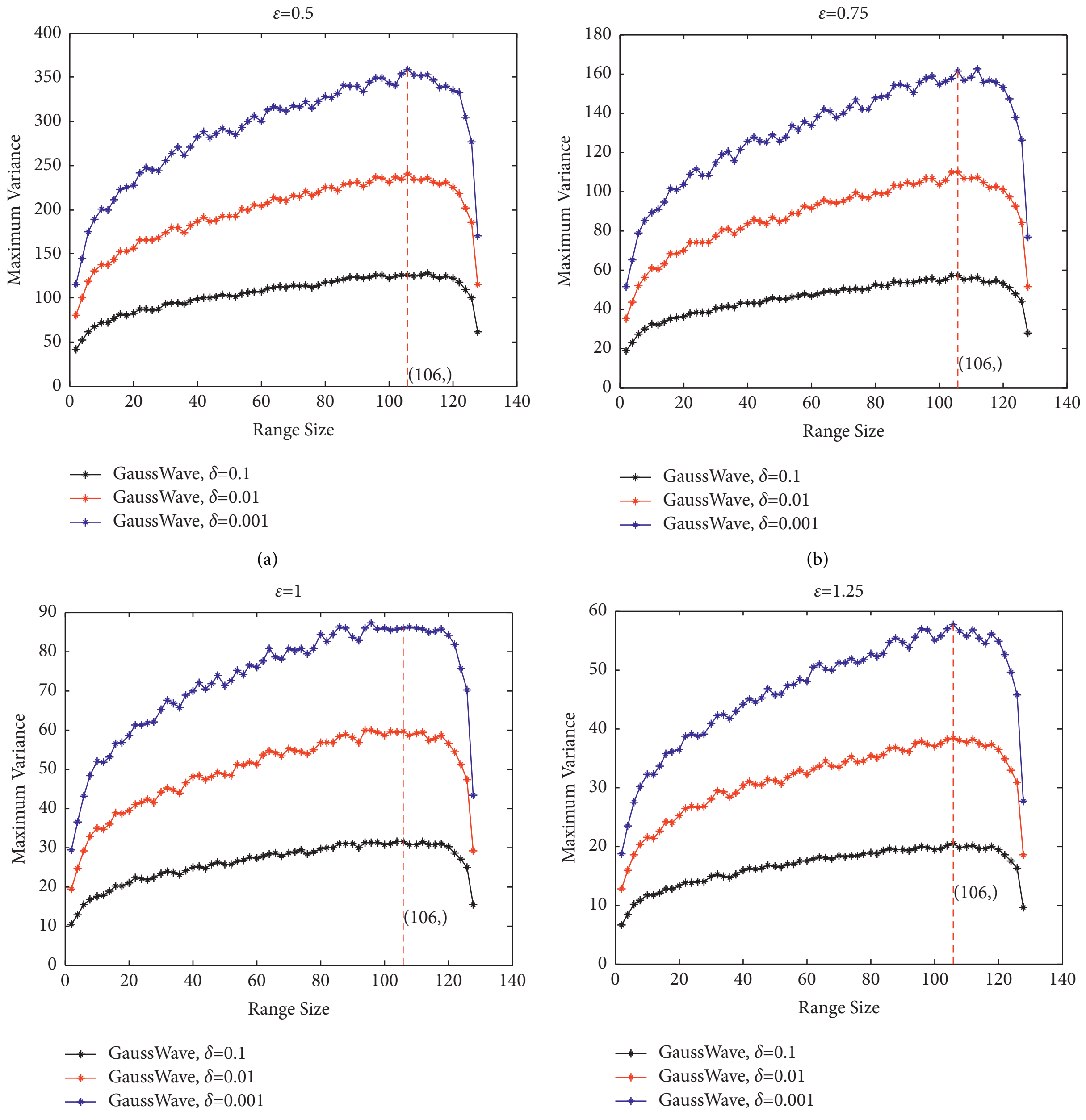
Figure 5: Continued.



(c)

(d)

Figure 5: Maximum variance of range query using Gaussian mechanism and Gaussian mechanism with Haar wavelet on age (the United States). (a) $\varepsilon = 0.5$. (b) $\varepsilon = 0.75$. (c) $\varepsilon = 1.0$. (d) $\varepsilon = 1.25$.



(a) (b) (c) (d)

Figure 6: Maximum variance of range query using Gaussian mechanism with Haar wavelet on age (the United States). (a) $\varepsilon = 0.5$. (b) $\varepsilon = 0.75$. (c) $\varepsilon = 1.0$. (d) $\varepsilon = 1.25$.

Table 9: Comparison of maximum value between experimental result and theoretical analysis for range 1 to 128.

ε	δ	Range size	Max-value	σ^2_{sumMax}	Difference
0.5	0.1	106	126.093366	126.252493	−0.159127
0.5	0.01	106	240.611130	241.347894	−0.736764
0.5	0.001	106	358.354318	356.451312	1.903006
0.75	0.1	106	56.957740	56.1109710	0.846769
0.75	0.01	106	109.807594	107.264005	2.543589
0.75	0.001	106	161.471355	158.420708	3.050647
1.0	0.1	106	31.521392	31.5617190	−0.040327
1.0	0.01	106	59.637942	60.336974	−0.699032
1.0	0.001	106	86.007548	89.112828	−3.105280
1.25	0.1	106	20.466279	20.199500	0.266779
1.25	0.01	106	38.351021	38.615663	−0.264642
1.25	0.001	106	57.678236	57.032210	0.646026

In Table 6, we observe that the max-variance of range query via Gaussian mechanism with Haar wavelet is $6.248291 * \sigma^2$ when $l = 7$ ($2^7 = 128$) and the length of interval is 106. Considering the value of σ in Table 8, we give the comparison of maximum value between experimental result and theoretical analysis for range query in Table 9.

In Table 9, the column "max-value" presents the experimental result of the maximum value of maximum variance for range query. The column "σ^2_{sumMax}" presents the result of theoretical analysis and computed using the formula $\sigma^2_{\text{sumMax}} = 6.248291 * \sigma^2$. The last column "difference" is the difference of columns "max-value" and "σ^2_{sumMax}." In Table 9, we find that the results of experiment and theoretical analysis are in substantial agreement.

This section gives the experimental verification of the framework of the maximum variance computing for range query. This framework on privacy preserving is built using Gaussian mechanism and Haar wavelet. In Figure 6, for any ε and δ, the maximum variance of the noise increases with the increase of range size before it gets the maximum value of 106, and it will decrease with the increase of range size after it has gotten the maximum value. In Table 9, the experimental value and the theoretical value of maximum variance for range count are compared, and the results show that they are in substantial agreement.

6. Conclusions

In this study, we proposed a new differential privacy framework via Haar wavelet transform and Gaussian mechanism for the range query. The theorems for how to inject Gaussian noise into the Haar wavelet coefficients are given. The noise of range query under the theoretical framework of Haar wavelet and Gaussian mechanism is analyzed. The algorithm to compute the maximum variance of any range query for any given parameter l is introduced. A coarse estimation of the maximum variance of range query using a function expression is given. The experimental results show that the maximum variance of the noise using Gaussian mechanism and Haar wavelet is far less than the noise using Gaussian mechanism. The experimental verification of the computing of maximum variance for range query based on lifting Haar wavelet and Gaussian mechanism is proposed, and the results show the experimental value and the theoretical value of maximum variance for range count are substantial agreement. For future work, we plan to apply our method to the privacy protection of histogram publication. Furthermore, we want to investigate how to assemble our method and machine learning algorithm, such as the decision tree and random forest.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Research Start-Up Funding Project of Qu Zhou University (nos. BSYJ202112 and BSYJ202109) and the Project for Public Interest Research Projects of Science and Technology Program of Zhejiang Province, China (nos. LGF21F010002, LGN21C130001, and LGG21F030002).

References

[1] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, *Preliminary of Differential Privacy," Differential Privacy and Applications*, pp. 7–16, Springer, Cham, 2017.

[2] P. Jain, M. Gyanchandani, and N. Khare, "Differential privacy: its technological prescriptive using big data," *Journal of Big Data*, vol. 5, no. 1, pp. 15–24, 2018.

[3] H. B. Kartal, X. Liu, and X. B. Li, "Differential privacy for the vast majority," *ACM Transactions on Management Information Systems (TMIS)*, vol. 10, no. 2, pp. 1–15, 2019.

[4] J. Lin, J. Niu, X. Liu, and M. Guizani, "Protecting your shopping preference with differential privacy," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1965–1978, 2021.

[5] H. Wang and H. Wang, "Correlated tuple data release via differential privacy," *Information Sciences*, vol. 560, pp. 347–369, Article ID 4267921, 2021.

[6] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and D. Megias, "Individual differential privacy: a utility-preserving formulation of differential privacy guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1418–1429, 2017.

[7] H. Wang and Z. Xu, "CTS-DP: publishing correlated time-series data via differential privacy," *Knowledge-Based Systems*, vol. 122, pp. 167–179, 2017.

[8] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 746–789, 2020.

[9] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1002–1012, 2020.

[10] X. Li, J. Yang, X. Sun, and Z. Jianpei, "Differential Privacy for Edge Weights in Social Networks," *Security and Communication Networks*, vol. 2017, pp. 1–10, 2017.

[11] P. Liu, Y. X. Xu, Q. Jiang et al., "Local differential privacy for social network publishing," *Neurocomputing*, vol. 391, pp. 273–279, 2020.

[12] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, and R. Wang, "Privacy-preserving approach PBCN in social network with differential privacy," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 931–945, 2020.

[13] X. Ren, C. M. Yu, W. Yu et al., "<tex-math notation=LaTeX>$\textsf{LoPub}$ </tex-math>: high-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[14] J. Wei, Y. Lin, X. Yao, and J. Zhang, "Differential privacy-based location protection in spatial crowdsourcing," *IEEE Transactions on Services Computing*, vol. 15, pp. 1–14, 2019.

[15] N. Almadhoun, E. Ayday, and Ö. Ulusoy, "Differential privacy under dependent tuples—the case of genomic privacy," *Bioinformatics*, vol. 36, no. 6, pp. 1696–1703, 2020.

[16] J. L. Raisaro, G. Choi, S. Pradervand et al., "Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1413–1426, 2018.

[17] M. Z. Hasan, M. S. R. Mahdi, M. N. Sadat, and N. Mohammed, "Secure count query on encrypted genomic data," *Journal of Biomedical Informatics*, vol. 81, pp. 41–52, 2018.

[18] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.

[19] L. Qian, T. Song, and A. Liang, "The optimization of the range query in differential privacy," in *Proceedings of the International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 1, pp. 618–623, IEEE, July 2015.

[20] J. Zhang, X. Xiao, and X. Xie, "Privtree: a differentially private algorithm for hierarchical decompositions," in *Proceedings of the 2016 International Conference on Management of Data*, pp. 155–170, San Francisco, CA, USA, January 2016.

[21] Z. Cai and Z. He, "Trading private range counting over big IoT data," *IEEE*, in *Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, July 2019.

[22] H. Mahdikhani, R. Lu, Y. Zheng, and A. Ghorbani, "Achieving efficient and privacy-preserving range query in fog-enhanced IoT with bloom filter," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, Dublin, Ireland, July 2020.

[23] H. Mahdikhani, R. Lu, J. Shao, and A. Ghorbani, "Using reduced paths to achieve efficient privacy-preserving range query in fog-based IoT," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4762–4774, 2021.

[24] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, pp. 797–822, 2013.

[25] X. Meng, H. Li, and J. Cui, "Different strategies for differentially private histogram publication," *Journal of Communications and Information Networks*, vol. 2, no. 3, pp. 68–77, 2017.

[26] Q. Han, B. Shao, L. Li, Z. Ma, H. Zhang, and X. Du, "Publishing histograms with outliers under data differential privacy," *Security and Communication Networks*, vol. 9, no. 14, pp. 2313–2322, 2016.

[27] H. Li, J. Cui, X. Meng, and J. Ma, "IHP: improving the utility in differential private histogram publication," *Distributed and Parallel Databases*, vol. 37, no. 4, pp. 721–750, 2019.

[28] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, *Differentially Private Data Publishing: Interactive Setting," Differential Privacy and Applications*, pp. 23–34, Springer, Cham, 2017.

[29] F. M. Bayer, A. J. Kozakevicius, and R. J. Cintra, "An iterative wavelet threshold for signal denoising," *Signal Processing*, vol. 162, pp. 10–20, 2019.

[30] M. Sharma, S. Patel, and U. R. Acharya, "Automated detection of abnormal EEG signals using localized wavelet filter banks," *Pattern Recognition Letters*, vol. 133, pp. 188–194, 2020.

[31] T. Suzuki, "Wavelet-based spectral–spatial transforms for CFA-sampled raw camera image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 433–444, 2020.

[32] Y. Zhang, "The fast image encryption algorithm based on lifting scheme and chaos," *Information Sciences*, vol. 520, pp. 177–194, 2020.

[33] S. P. Singh and G. Bhatnagar, "A simplified watermarking algorithm based on lifting wavelet transform," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 20765–20786, 2019.

[34] M. B. Nagare, B. D. Patil, and R. S. Holambe, "On the design of biorthogonal halfband filterbanks with almost tight rational coefficients," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 4, pp. 790–794, 2020.

[35] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1200–1214, 2011.

[36] P. Derbeko, S. Dolev, and E. Gudes, "Wavelet-based dynamic and privacy-preserving similitude data models for edge computing," *Wireless Networks*, vol. 27, no. 1, pp. 351–366, 2021.

[37] C. Lin, P. Wang, H. Song, Y. Zhou, Q. Liu, and G. Wu, "A differential privacy protection scheme for sensitive big data in body sensor networks," *Annals of Telecommunications*, vol. 71, no. 9-10, pp. 465–475, 2016.

[38] C. Dwork, K. Talwar, A. Thakurta, and Li. Zhang, "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 11–20, May 2014.

[39] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, pp. 211–407, 2013.

[40] S. Ruggles, K. Genadek, and R. Goeken, *Integrated public use microdata series: Version 6.0 [dataset]*, University of Minnesota, vol. 10, p. 010, Minneapolis, 2017.