cryptography

MDPI

Article

Blockchain of Resource-Efficient Anonymity Protection with Watermarking for IoT Big Data Market

Chia-Hui Wang [1,*] and Chih-Hao Hsu [2]

[1] Department of Computer Science and Information Engineering, Ming Chuan University, No. 5, De Ming Road, Taoyuan 33348, Taiwan
[2] Department of Information Management, Ming Chuan University, No. 5, De Ming Road, Taoyuan 33348, Taiwan
* Correspondence: wangch@mail.mcu.edu.tw

Abstract: According to the ever-growing supply and demand of IoT content, IoT big data in diversified applications are deemed a valuable asset by private and public sectors. Their privacy protection has been a hot research topic. Inspired by previous work on bounded-error-pruned IoT content market, we observe that the anonymity protection with robust watermarking can be developed by further pruning data for better resource-efficient IoT big data without violating the required quality of sensor service or quality of decision-making. In this paper, resource-efficient anonymity protection with watermarking is thus proposed for data consumers and owners of IoT big data market via blockchain. Our proposed scheme can provide the IoT data with privacy protections of both anonymity and ownership in IoT big data market with resource efficiency. The experiments of four different-type IoT datasets with different settings included bounded-errors, sub-stream sizes, watermark lengths, and ratios of data tampering. The performance results demonstrated that our proposed scheme can provide data owners and consumers with ownership and anonymity via watermarking the IoT big data streams for lossless compressibility. Meanwhile, the developed DApp with our proposed scheme on the Ethereum blockchain can help data owners freely share and trade with consumers in convenience with availability, reliability, and security without mutual trust.

Keywords: anonymity; watermark; blockchain; IoT big data market

Citation: Wang, C.-H.; Hsu, C.-H. Blockchain of Resource-Efficient Anonymity Protection with Watermarking for IoT Big Data Market. *Cryptography* **2022**, *6*, 49. https://doi.org/10.3390/cryptography6040049

Academic Editor: Cheng-Chi Lee

Received: 28 August 2022
Accepted: 26 September 2022
Published: 30 September 2022

1. Introduction

On the Internet of Things (IoT), technology and diverse applications where everything can be connected to the Internet, IoT systems can automatically sense, collect, display, or transmit information for a long time. These applications bring about a close relationship between human life and intelligence technologies. In recent years, due to continuous breakthroughs in communication technology with cloud, fog, and edge computing technology, the speed of large amounts of IoT data generated is accelerated by the faster-than-ever networking performance.

For the data owners in IoT big data markets [1,2], whether individuals, enterprises, or government units, their diversified IoT data are regarded as very important assets. However, most of the cloud storage services for these IoT big data are provided by third-party service providers. It may have risks of data instability, leakage, and vulnerability. Efficient privacy protection schemes for IoT big data have been a very popular research topic. Further, governments in various countries have paid much more attention to the protection of personal data privacy than before by making relevant laws and regulations. They hope to balance data privacy and data availability without excessively restricting data usability, such as the General Data Protection Regulations (GDPR) [3] set by the European Union, which strictly regulates the personal data privacy.

Data anonymity is information with the intention of data privacy protection by removing personally identifiable information in datasets, so that the identification that the

data describe remains anonymous. Due to the diversified IoT applications usually having their own bounded-error tolerance in their specific IoT data, in this paper, we extend the idea of privacy protection for the bounded-error-pruned (BEP) IoT content marketplace (BIoTCM) [4]. The extended idea is that the degree of IoT data anonymity can be decided by the bounded-errors tolerated by data consumers for their IoT applications. That is to say, the larger bounded-error indicates not only higher compression ratios for the BEP IoT data in resource efficiency, but also a higher degree of data anonymity for data consumers using BEP IoT data.

Moreover, we further take the advantage of the error-bounded interval ranging from low-bound to upper-bound in data owners' BEP IoT data, and a simple watermarking scheme within the assigned bounded-error in BEP IoT data for owners' ownership is proposed. To the best of our knowledge, since the proposed watermarking scheme does not affect the bounded-error tolerance in diversified IoT applications, our proposed watermarking scheme is a kind of zero-bit watermarking [5–7] for ownership of BEP IoT data in the IoT big data market. For the IoT big data market, we furnish this resource-efficient anonymity protection with the watermarking (RAPW) scheme for data consumers' anonymity and owners' ownership.

The proposed architecture of the IoT big data market with the RAPW scheme via an Ethereum blockchain [8] is shown in Figure 1. We designed and developed a DApp for an anonymous bounded-error IoT big data marketplace to assist IoT big data owners and consumers without trusting each other. The IoT big data sharing and transactions with data reliability and security were carried out through Ethereum smart contracts. The owners' IoT data source including online data from remote sensors and offline data from databases can be linked to this DApp. Owners can further set different bounded-errors (i.e., data resolutions) and watermarks through DApp to the backend blockchain system for generating their RAPW-BEP IoT data on the decentralized IPFS file system. Then, these RAPW-BEP IoT data with anonymity and watermarks can be purchased and downloaded by consumers. The RAPW-BEP IoT data streams in owner-to-consumer pairs can still preserve the energy efficiency in online sensing data and offline IoT big data.

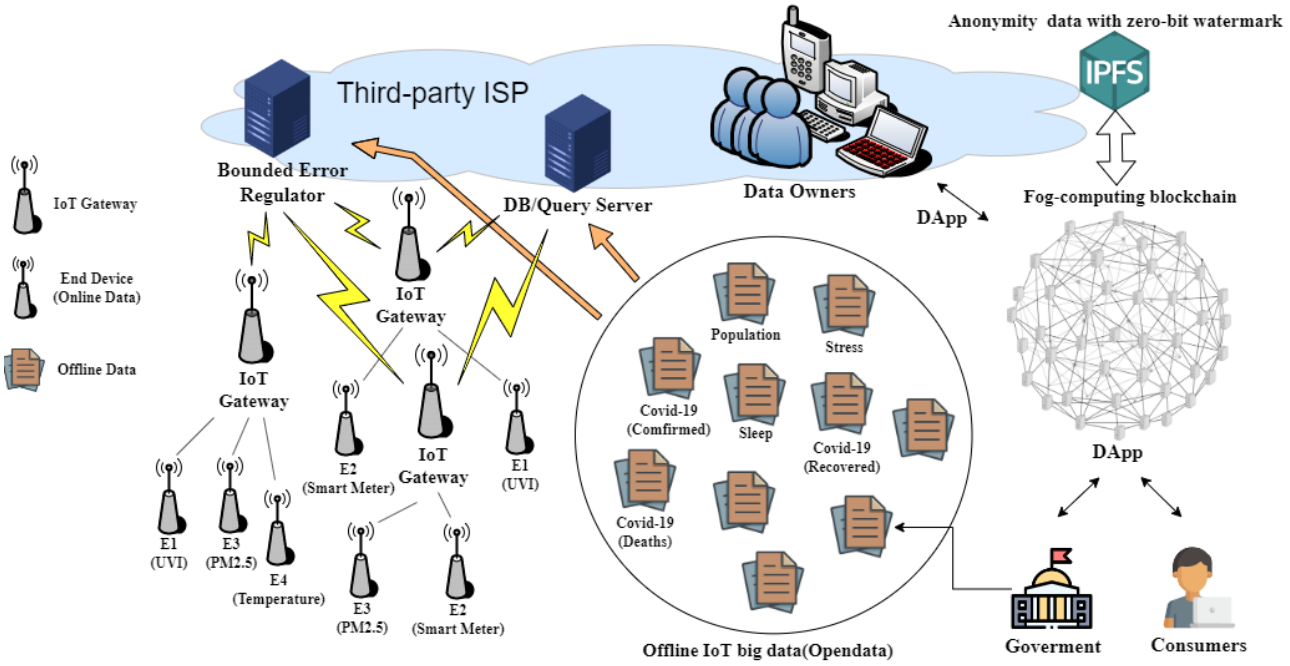

Figure 1. Proposed architecture of IoT big data market via Ethereum blockchain.

This paper conducted a performance test for the proposed RAPW scheme using different kinds of real-world IoT datasets including temperature, the UV index, and COVID-19 open data. The preliminary results of the experiment show that the anonymous bounded-

error IoT big data market using the RAPW scheme on an Ethereum blockchain can successfully generate the specified BEP data that conforms to bounded distortions of the original data and ensure good accuracy of watermark extraction even from data tampering. It is believed that such an architecture of the IoT big data market with the proposed RAPW scheme can encourage more IoT data owners including government departments to provide data consumers with valuable data for more diversified IoT applications.

2. Related Works

In diversified IoT systems, due to their applied technologies or hardware barriers, the sensing IoT big data often differs from the real values such as analog to digital or environmental noise. However, this error should be bounded to avoid excessive errors to jeopardize the credibility of their IoT applications for users. The compression schemes for resource efficiency in IoT data by taking advantage of the bounded-error can be classified as lossy compression [9–11] and lossless compression [4,12]. The error-bounded lossy compression schemes usually have higher compression performance than error-bounded lossless compression schemes. Since error-bounded lossy compression schemes cannot recover the original data from compressed data, they must take extra efforts to recover the error-bounded data close to the original data.

As the requirements from the quality of sensor service (QoS^2) and the quality of decision (QoD) in respective online and offline IoT big data usually have bounded-error tolerance in a real IoT system, the previous research proposed the LBE-RLE (layered bounded-error run-length-encoding) compression scheme [12], as illustrated in Figure 2, for online sensor data to reduce the power consumption of IoT communication to extend the IoT system lifetime.

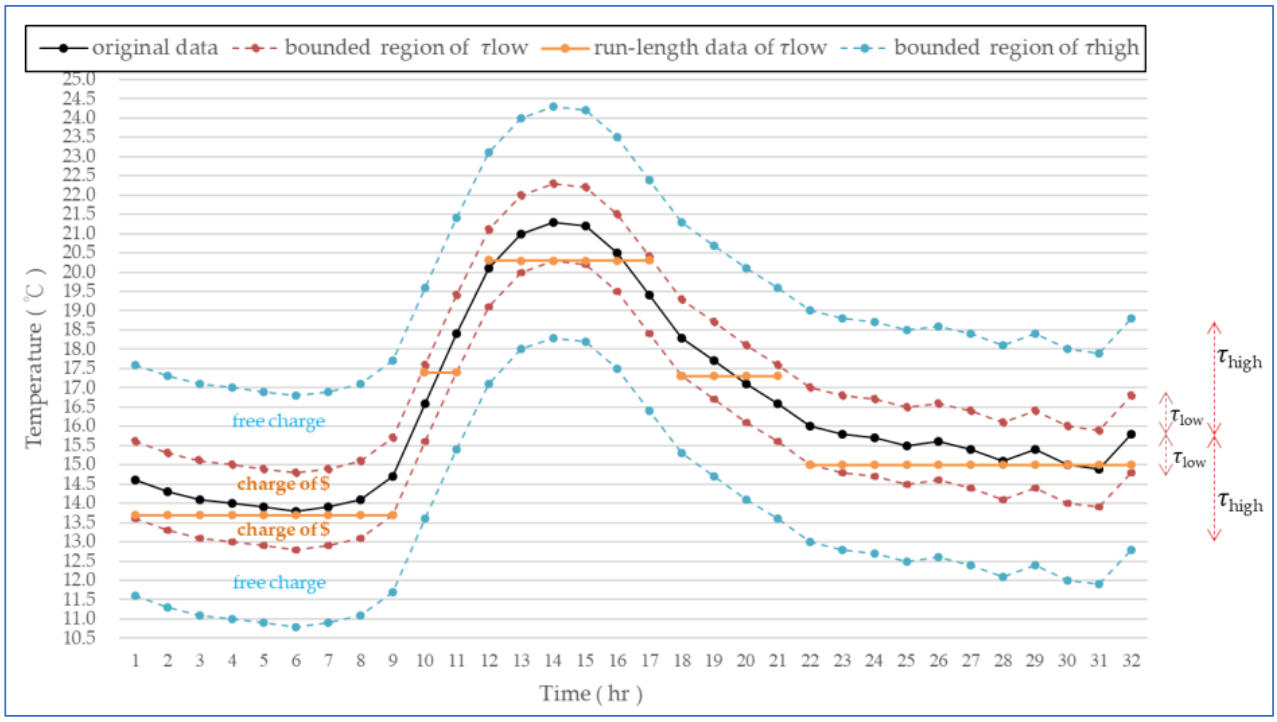

Figure 2. A temperature sub-stream using LBE-RLE compression for resource-efficient IoT data [4].

In Figure 2, according to the assigned bounded-error (i.e., $\tau_{low} = 1$), in the 32 samples of a temperature data stream in the black curve, this heuristic LBE-RLE lossless compression scheme transforms the original data stream <14.6, 14.3, 14.1, 14, 13.9, 13.8, 13.9, 14.1, 14.7, 16.6, 18.4 20.1, 21, 21.3, 21.2, 20.5, 19.4, 18.3, 17.7, 17.1, 16.6, 16, 15.8, 15.7, 15.5, 15.6, 15.4, 15.1,15.4, 15, 14.9, 15.8> into orange line segments of the run-length sequence <[13.7, 9], [17.4, 2], [20.3, 6], [17.3, 4], [15, 11]> starting at the lower-bound value of 13.7

(i.e., $14.6 - \tau_{\text{low}} + 0.1$) with much better compression performance for resource-efficient transmission and storage of bounded-error-pruned (BEP) IoT temporal data without jeopardizing IoT application's QoS^2/QoD.

Moreover, taking advantage of the BEP sensor data from different bounded errors can provide different consumers with different QoS^2/QoD requirements in their own IoT applications. The previous work [4] proposed the above-mentioned BIoTCM framework for owners and consumers on an Ethereum blockchain network to have preliminary privacy protection for their BEP data with different market values. As shown in Figure 2, the BEP data stream with a higher-bounded-error (i.e., τ_{high}) usually preserves a lower price or is even free of charge in the proposed BIoTCM. That is to say, the BEP data stream with the lower-bounded-error τ_{high} usually has a higher price for consumers, because its stream values are much closer to the original data stream (i.e., the black curve). Through the BIoTCM Ethereum smart contract [13], the owner's original data stream and its BEP data stream with different bounded-errors are transferred to the reliable IPFS (interplanetary file system [14]). The files on the P2P IPFS file system preserve unique hash values for different file contents [15]. Thus, these hash values can be used to retrieve these corresponding files respectively for content integrity. Meanwhile, BIoTCM can simply protect the privacy of different groups of consumers requesting the different BEP data streams with different BE resolutions in the owner's original data. For further privacy protection between consumers requesting the same BEP data stream, BIoTCM stores the different PKI-encrypted files in LBE-RLE compression for different consumers requesting the same BEP data streams.

In the literature review of [16], they identified wide-ranging and creative methodologies for cyber analytics and explored the risks of deliberately influencing or disrupting behaviors in sociotechnical systems fostered by IoT big data. They argued that the design of IoT big data systems for edge computing environments is challenging, and one of the most pressing points is security. In the significant research topic of multimedia processing, digital watermarking is an imperceptible change to original digital media for the ownership provision against piracy [17]. In this paper, we further proposed resource-efficient anonymity protection with watermarking (RAPW) for IoT big data market via Ethereum blockchain technology. Both the anonymity for different consumers and the watermarking ownership for owners are resource-efficiently preserved in the IoT big data market, because LBE-RLE-compressed files in smaller sizes for BEP data streams are stored on the P2P IPFS file system through the DApp [18] and smart-contract deployed on the Ethereum blockchain.

3. System Architecture and Proposed Scheme

3.1. Preliminaries for Resource Efficiency in BEP IoT Big Data

In the previously proposed LBE-RLE (layered bounded-error run-length-encoding) scheme [12], the original sensor data from n samples can be represented in a data stream $D_{sensor}(n)D_{sensor}(n)$ as defined in Equation (1). In Equation (2), the BEP data stream $D_{LBE}(\tau)$, which has sequences of the same data values by the assigned bounded-error of τ, can be further represented as $D_{LBE-RLE}(\tau)$ for LBE-RLE lossless compression as shown in Equation (3). The RLE subsequence of $\left[\widetilde{d}_i^{\tau}, r_i\right]$ with r_i repeated data value $\widetilde{d}_i^{\tau}$ is defined in Equation (4). However, the RLE subsequence $\left[\widetilde{d}_i^{\tau}, 1\right]$ (i.e., $r_i = 1$) is usually encoded as $\widetilde{d}_i^{\tau}$ without coding the run-length of 1 as defined in Equation (5).

$$D_{sensor}(n) \equiv \langle d_1, d_2, \cdots, d_n \rangle \tag{1}$$

$$D_{LBE}(\tau) \equiv \langle \widetilde{d}_1^{\tau}, \ldots, \widetilde{d}_1^{\tau}, \widetilde{d}_2^{\tau}, \cdots, \widetilde{d}_2^{\tau}, \ldots, \widetilde{d}_m^{\tau}, \ldots, \widetilde{d}_m^{\tau} \rangle,\ m \le n \tag{2}$$

$$D_{LBE-RLE}(\tau) \equiv \langle \left[\widetilde{d}_1^{\tau},\ r_1\right], \left[\widetilde{d}_2^{\tau}, r_2\right], \cdots, \left[\widetilde{d}_m^{\tau}, r_{\mathrm{m}}\right] \rangle,\ r_i > 1,\ m \le n = \textstyle\sum_{i=1}^{m} r_i \tag{3}$$

$$\left[\widetilde{d}_i^{\tau}, r_i\right] \equiv \cup_{r_i} \langle \widetilde{d}_i^{\tau} \rangle,\ \exists \widetilde{d}_i^{\tau} : \forall_j \left| d_j - \widetilde{d}_i^{\tau} \right| \le \tau, \left(\textstyle\sum_{k=1}^{i-1} r_k\right) + 1 \le j \le \textstyle\sum_{k=1}^{i} r_k, r_0 = 0 \tag{4}$$

$$\left[\widetilde{d}_i^{\tau}, r_i\right] \equiv \widetilde{d}_i^{\tau},\ if\ r_i = 1 \tag{5}$$

To theoretically demonstrate the resource efficiency in BEP IoT big data while using lossless compression (i.e., Huffman coding), the average coding bit length of Shannon entropy for original data samples $D_{sensor}(n)$ and LBE-RLE samples $D_{LBE-RLE}(\tau)$ is utilized, as shown in Equations (6) and (7), respectively. They indicate the lower bounds of mean coding length [19]. The BEP data stream $D_{LBE-RLE}(\tau)$ in Equation (7) has more redundancy in $\widetilde{d}_i^{\tau}$ than the original sensor data stream $D_{sensor}(n)$ in Equation (6). Thus, the BEP data stream $D_{LBE-RLE}(\tau)$ has better RLE lossless compression in resource efficiency compared to the original sensor data stream $D_{sensor}(n)$. In Equations (6) and (7), the function $p(x)$ represents the occurrence probability of data value x in its data stream. Thus, the BEP-enabled IoT sensors can save more power to transmit the compressed data for extending the system lifetime without sacrificing QoS^2/QoD requirements in diversified IoT applications.

$$entropy(D_{sensor}(n)) = \sum_{i=1}^{n} \log_2 \frac{1}{p(d_i)} p(d_i) \tag{6}$$

$$entropy(D_{LBE-RLE}(\tau)) = \sum_{i=1}^{m} \log_2 \frac{1}{p\left(\widetilde{d}_i^{\tau}\right)} p(\widetilde{d}_i^{\tau}) + \sum_{i=1}^{m} \log_2 \frac{1}{p(r_i)} p(r_i), m \leq n = \sum_{i=1}^{m} r_i \tag{7}$$

In the proposed IoT big data content market, the original IoT data of a data owner can be represented as Equation (8), similar to Equation (1). Then the consumer can buy different BEP data $D_{LBE-RLE}^{Owner}(\tau_j)$, as defined in Equation (9), from this data owner with a different bounded-error τ_j in the original data $D_{data}^{Owner}(n)$. Since the sequences of the same data values, as defined in Equation (9), are found under the condition of $\forall_{\overline{J}}^{=} \left| d_{\overline{J}}^{owner} - \widetilde{d}_i^{\tau_j(owner)} \right| \leq \tau_j$ between the original data value $d_{\overline{J}}^{owner}$ and its BEP value $\widetilde{d}_i^{\tau_j(owner)}$, the different τ_j applied in Equation (10) will have different BEP data sequences as defined in Equation (9) to achieve different data compressions. That is to say, the data streams $D_{LBE-RLE}^{Owner}(\tau_j)$ for consumers requesting the owner's BEP data with same error bound τ_j are all the same to preserve the anonymity between these consumers. On the other hand, privacy is preserved for different groups of consumers requesting the owner's BEP data with different error bounds τ_j (e.g., $D_{LBE-RLE}^{Owner}(0.5) \neq D_{LBE-RLE}^{Owner}(1.0)$).

$$D_{data}^{Owner}(n) \equiv \langle {d_1}^{owner}, {d_2}^{owner}, \cdots, {d_n}^{owner} \rangle \tag{8}$$

$$D_{LBE-RLE}^{Owner}(\tau_j) \equiv \langle \left[\widetilde{d}_1^{\tau_j(owner)},\ r_1\right], \left[\widetilde{d}_2^{\tau_j(owner)}, r_2\right], \cdots, \left[\widetilde{d}_m^{\tau_j(owner)}, r_{\mathrm{m}}\right] \rangle,\ r_i > 1,\ m \leq n = \sum_{i=1}^{m} r_i \tag{9}$$

$$\left[\widetilde{d}_i^{\tau_j(owner)}, r_i\right] \equiv \cup_{r_i} \langle \widetilde{d}_i^{\tau_j \tau_j(owner)} \rangle,\ \exists \widetilde{d}_i^{\tau_j(owner)} : \forall_{\overline{J}}^{=} \left| d_{\overline{J}}^{owner} - \widetilde{d}_i^{\tau_j(owner)} \right| \leq \tau_j, \left(\sum_{k=1}^{i-1} r_k\right) + 1 \leq \overline{J} \leq \sum_{k=1}^{i} r_k, r_0 = 0 \tag{10}$$

$$\left[\widetilde{d}_i^{\tau_j(owner)}, r_i\right] \equiv \widetilde{d}_i^{\tau_j(owner)},\ if\ r_i = 1 \tag{11}$$

3.2. System Architecture for IoT Big Data Market Using Ethereum Blockchain

In our system architecture for the IoT big data market (Figure 1), the concept of BEP IoT content was used to not only extend the online IoT system lifetime [12,20], but also to provide resource-efficient anonymity protection with watermark (RAPW) for IoT big data via the blockchain smart contract. All the original data and its RAPW-BEP data for data owners and consumers are lossless-compressed and then saved to a P2P IPFS file for resource efficiency. The returned hash values for these IPFS files are stored in the blockchain via a smart contract for later file retrieval with integrity. Since the hash value varies if the

IPFS file is contaminated, the blockchain transaction information of the hash values for IPFS files guarantee the content integrity of RAP-BEP-compressed files for IoT big data.

The backend smart contract provides data consumers with the requested IoT content in different BEP precision from user-friendly frontend DApp. Then, the proposed RAPW scheme can be used to provide IoT big data content in the marketplace for different-level consumers' access, according to the authority and payment between data consumers and owners. In the RAPW scheme, the proposed RAPW-BEP algorithm uses the data owner's original IoT data stream to generate different BEP data streams with ownership watermarks for different-level data consumers' requests in bounded-error criteria of their own IoT applications. Using our proposed RAPW-BEP algorithm, different owners can insert their data ownership watermarks via the DApp to their different BEP data streams for consumers. Meanwhile, the ownership of the RAPW-BEP data streams in the IoT big data marketplace can be validated via our proposed RAPW-ReadWM algorithm, even if the RAPW-BEP data stream has been contaminated to a certain degree.

However, the different bounded-errors assigned for different RAPW-BEP data streams with the same watermark indicate the different degree of anonymity for the owner's original IoT data. That is to say, data consumers may request the same RAPW-BEP data stream with the same BE and watermark. Then, only one single RAPW-RLE-compressed file is stored on the P2P IPFS file system. To further protect the privacy of these consumers using the same BE and watermark, the PKI-encrypted IPFS files for consumers who require further privacy protection can be generated accordingly in spending more resources on P2P IPFS systems in the IoT big data market.

The above-mentioned system flowchart of the proposed RAPW scheme using blockchain technology for the IoT big data market is summarized in the Figure 3. More detail of algorithms RAPW-RLE, RAPW-BEP, and RAPW-ReadWM in the proposed RAPW scheme for the resource-efficient IoT big data market's privacy protection in anonymity with watermarking is described in the following sub-section.

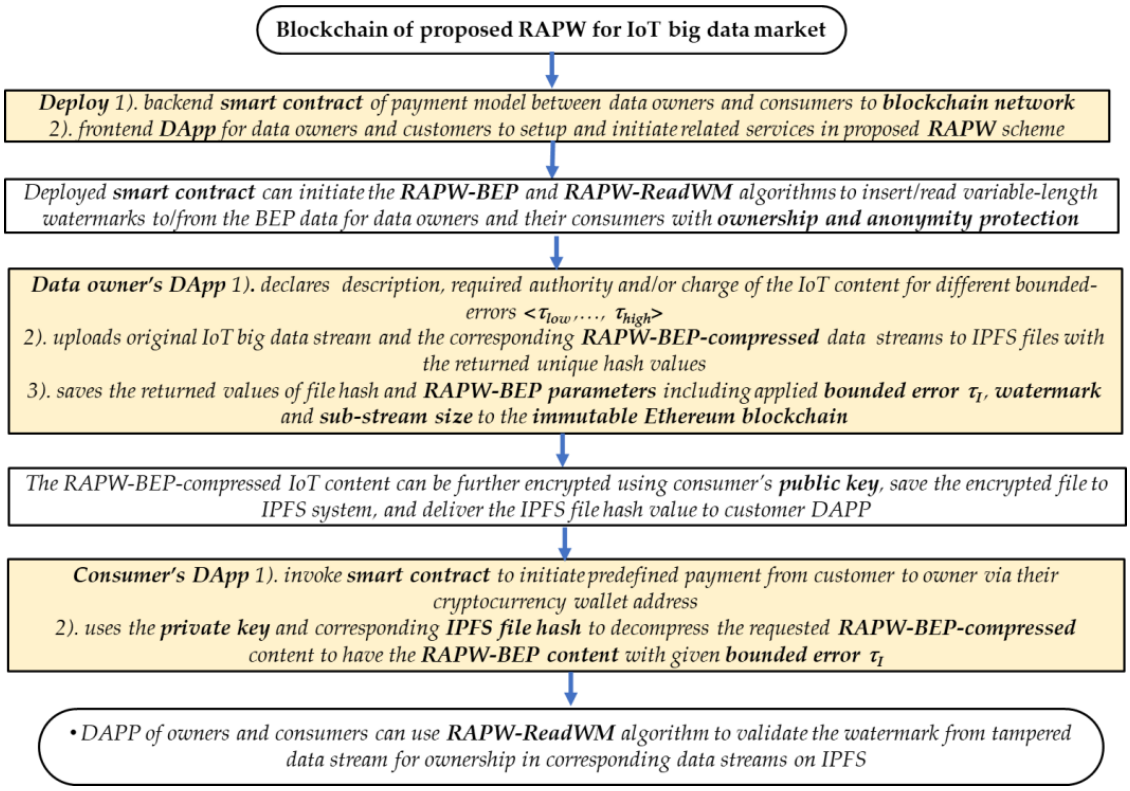

Figure 3. System flowchart of RAPW-enabled blockchain for IoT big data market.

3.3. Resource-Efficient Anonymity Protection with Watermark (RAPW) Scheme

In the previous LBE-RLE example in Figure 2, the bounded-error τ can be regarded as the anonymity degree of BEP data to the original data. The higher-bounded τ preserved in the BEP data stream indicates that the higher anonymity with a higher compression ratio for IoT big data market can be resource-efficiently achieved. In our proposed resource-efficient anonymity protection with a watermark (RAPW) scheme, the starting point $\widetilde{d}_1^{\tau}$ of the BEP data stream is not heuristic anymore. In the proposed RAPW-RLE scheme, the starting point is decided by the watermark inserted into the BEP data stream within the constraint of bounded-error τ. As shown in Figure 4, the zooming view of Figure 2 indicates that there are 19 possible starting points to achieve different LBE-RLE encoding (e.g., the orange, green and blue lines), after we excluded the upper-bound and lower-bound curves in the bounded region of the black curve within the bounded-error $\tau = 1$. Thus, we redefined the original IoT big data stream into N sub-stream S_i as shown in Equation (12). The sub-stream S_i is then defined in Equation (13). Each owner's watermark of its ownership can be encoded to an integer stream $W(l)$ of l integers, which is less than 20 or 10, as shown in Equations (14) and (15) for bounded errors of $\tau = 1$ and $\tau = 0.5$, respectively.

$$D_{Bigdata}(N) \equiv \langle S_1, S_2, \cdots, S_N \rangle \tag{12}$$

$$S_I(n_I) \equiv \langle d_1^I, d_2^I, \cdots, d_{n_I}^I \rangle,\ I = 1, N \tag{13}$$

$$W(l) \equiv \langle w_1, w_2, \cdots, w_l, w_{delimiter} \rangle,\ 1 \le w_k \le 19,\ \tau = 1,\ 1 \le l \ll N \tag{14}$$

$$W(l) \equiv \langle w_1, w_2, \cdots, w_l, w_{delimiter} \rangle,\ 1 \le w_k \le 9,\ \tau = 0.5,\ 1 \le l \ll N \tag{15}$$

$$S_I^{RAPW-RLE}(\tau, w_k) \equiv \langle \left[\widetilde{d}_{1,w_k}^{\tau}, \widetilde{r}_1\right], \left[\widetilde{d}_{2,w_k}^{\tau}, \widetilde{r}_2\right], \cdots, \left[\widetilde{d}_{m,w_k}^{\tau}, \widetilde{r}_m\right] \rangle,\ m \le n_i = \sum_{\rightarrow i=1}^{m} \widetilde{r}_{\rightarrow i} \tag{16}$$

$$\widetilde{d}_{i,w_k}^{\tau} \equiv d_j^I - \tau + w_k \times 0.1, j = \left(\sum_{\rightarrow i=1}^{i-1} \widetilde{r}_{\rightarrow i}\right) + 1 \tag{17}$$

$$\therefore w_k = \widetilde{d}_{i,w_k}^{\tau} - (d_j^I - \tau) \Big/ 0.1, j = \left(\sum_{\rightarrow i=1}^{i-1} \widetilde{r}_{\rightarrow i}\right) + 1 \tag{18}$$

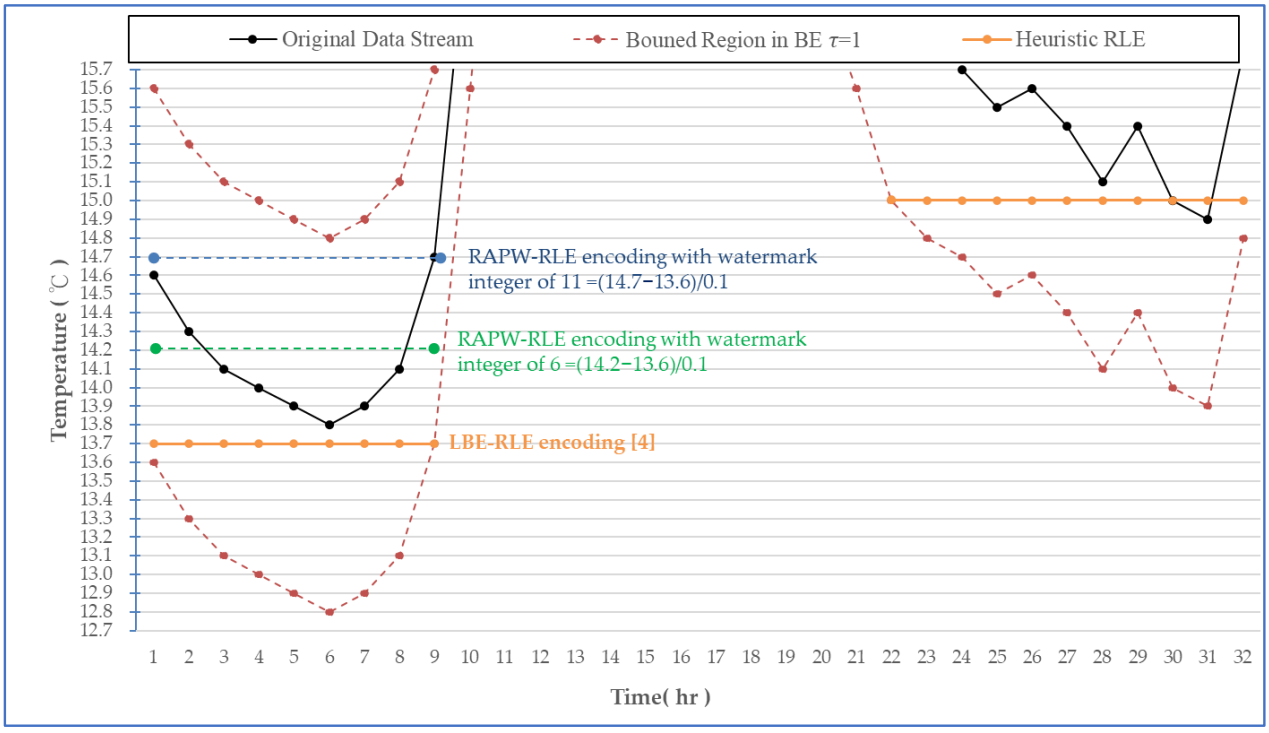

Figure 4. Two RAPW-RLE sub-stream samples using two different watermark integers.

Without the loss of generality, as shown in Figures 2 and 4, the bounded region around the original data can be evenly divided into 20 or 10 tracks, if τ = 1 or τ = 0.5, which are $\{d_1^I - \tau + 0.1 \times w_j,\ 1 \leq w_j \leq 19\}$ or $\{d_1^I - \tau + 0.1 \times w_j, 1 \leq w_j \leq 9\}$ in the first data of original data sub-stream $S_1(n_1)$. Thus, the l + 1 integers (including $w_{delimiter}$) from encoded watermark $W(l)$ can be sequentially inserted into the first l + 1 sub-streams (i.e., $S_1, S_2, \cdots, S_l, S_{l+1}$) by sequentially performing the RAPW-RLE algorithm, as shown Algorithm 1, to have the watermarked sub-streams (i.e., $S_1^{RAPW-RLE}$, $S_2^{RAPW-RLE}$, $\ldots$, $S_l^{RAPW-RLE}$) with BEP data for LBE-RLE compression in resource-efficient IoT big data market. Since l is assumed much less than the total number N of sub-streams in original data stream $D_{Bigdata}(N)$ defined in Equation (12), the l-integer-encoded watermark $W(l)$ can be repeatedly inserted at $floor(N/(l+1))$ times, because the delimiter $w_{delimiter}$ is needed to correctly read the l-integer-encoded watermark out from the N RAPW-RLE sub-streams (i.e., $S_I^{RAPW-RLE}(\tau, w_k)$ in Equation (16)), which has been RAPW-RLE-encoded from original IoT big data stream $D_{Bigdata}(N)$ defined in Equation (12).

In the RAPW-RLE algorithm for RAPW-RLE-encoding the sub-streams $S_I(n_I)$, as defined in Equation (13) of original IoT big data, we can use the boundary index w_k of the lower-bound (e.g., $d_1^I - \tau$) or the upper-bound (e.g., $d_1^I + \tau$) in the τ-bounded region as the delimiter $w_{delimiter}$ (i.e., $w_{delimiter}$= 0, or $w_{delimiter}$ = 10 for τ = 0.5 and $w_{delimiter}$ = 20 for τ = 1), since the initial data in RAPW-RLE-encoding starting at the boundary line may have worse LBE-RLE encoding performance for compression without longer running BEP data, as shown in Equations (6) and (7).

In Algorithm 2, we provide the RAPW-BEP algorithm to insert the sequential watermark integers $W(l)$ as defined in Equation (14) or Equation (15) into the N RAPW-RLE-encoding sub-streams $S_I^{RAPW-RLE}(\tau, w_k)$ repeatedly for original IoT big data stream $D_{Bigdata}(N)$.

As with the description in the previous Section 3.2, the owner's original IoT data are RAPW-BEP-encoded via algorithms of RAPW-BEP and RAPW-RLE. Then, the RAPW-BEP-encoded data stream is compressed and saved into P2P IPFS files. While the RAPW-BEP-compressed file from the IPFS file system is needed to later validate the ownership, the ownership watermark inserted by RAPW-BEP and RAPW-RLE algorithms can be read out correctly via the RAPW-ReadWM algorithm as described in Algorithm 3.

According to the RAPW-RLE algorithm, each watermark integer in the watermark integer string is repeatedly inserted into the starting data value $\widetilde{d}_{i,w_k}^{\tau}$ of every RLE run of $\left[\widetilde{d}_{i,w_k}^{\tau}, \widetilde{r}_i\right]$ in the RAPW-RLE-encoded sub-stream, as defined in Equation (17). Then, our proposed RAPW-ReadWM algorithm can read out the watermark and allows some noise in the RAPW-RLE-encoded sub-stream within a certain degree via a given noise threshold. Since the RAPW-ReadWM algorithm counts the occurrences of possible watermark integers as calculated in Equation (18) in each run of $\left[\widetilde{d}_{i,w_k}^{\tau}, \widetilde{r}_i\right]$. If the total of the occurrence values in different watermark integers other than the watermark integer with maximum occurrence is no more than a given threshold, the watermark integer with maximum occurrence is confirmed as a legal watermark integer for this RAPW-RLE-encoded sub-stream.

Thus, the proposed RAPW scheme can provide IoT big data market with resource-efficient anonymity and watermark protection. The watermark protection with robustness can be realized by the noise threshold (i.e., *wrThreshold*) given in the RAPW-ReadWM algorithm as shown in Algorithm 3.

Algorithm 1. RAPW-RLE algorithm for a IoT data sub-stream with single watermark integer w_k.

Input: bounded-error $\boldsymbol{\tau}$, sub-stream $\boldsymbol{S_i}$ of IoT big data, integer digit $\boldsymbol{w_j}$ of watermark $\{w_1, w_2, \ldots, w_l, w_{\text{delimiter}}\}$

Output: RLE sub-stream $\boldsymbol{S_i^{RAPW-RLE}}$ for τ and w_j

1: n = $\boldsymbol{S_i}$.dataSize
2: upperbound_S_i[1:n].data = $\boldsymbol{S_i}$[1:n].data + $\boldsymbol{\tau}$
3: lowerbound_S_i[1:n].data = $\boldsymbol{S_i}$[1:n].data − $\boldsymbol{\tau}$
4: Startindex = 1
5: Endindex = n
6: runLength = 1
7: **while** (Startindex < n + 1)
8: startingData = lowerbound_S_i[Startindex].data + $\boldsymbol{w_j}$ ×0.1
9: **while** (Endindex > Startindex)
10: **if** (startingData > upperbound_S_i[Endindex].data) **or** (startingData < lowerbound_ S_i[Endindex].data))
11: runLength = 1
12: **else**
13: runLength ++
14: **end if**
15: Endindex--
16: **end while**
19: $\boldsymbol{S_i^{RAPW-RLE}}$.append(startingData)
17: **if** (runLength > 1)
18: $\boldsymbol{S_i^{RAPW-RLE}}$.append(runLength)
20: Startindex = Startindex + runLength
21: **end while**

Algorithm 2. RAPW-BEP algorithm for IoT data stream with watermark integer string.

Input: bounded-error $\boldsymbol{\tau}$, IoT big data stream $\boldsymbol{D_{Bigdata}}$, maximum size ***sizeMax*** for sub-stream of $\boldsymbol{D_{Bigdata}}$, owner's watermark $\boldsymbol{W}$[1:l + 1] = $\{w_1, w_2, \ldots, w_l, w_{\text{delimiter}}\}$

Output: RAPW-RLE data stream $\boldsymbol{D_{RAPW-RLE}}$ with watermark $W(l)$

1: N = $D_{Bigdata}$.dataSize
2: sizeRemained = N
3: I = j = 1
4: **while** (sizeRemained > ***sizeMax***)
5: S_i.data = $\boldsymbol{D_{Bigdata}}$[(i − 1) × ***sizeMax*** + 1:i×***sizeMax***]
6: S_i.dataSize = ***sizeMax***
7: w_j **= W[j]**
8: $\boldsymbol{D_{RAPW-RLE}}$.append(**RAPW-RLE(**$\boldsymbol{\tau}$, S_i, w_j**)**) /* in Algorithm 1 */
9: j = j + 1
10: I = I + 1
11: **if** ($j \geq \boldsymbol{l}$ + 1) /* ***l*** is defined in ***W*[1:*l*]** in Equation (14) or (15) */
12: j = 1
13: **end if**
14: sizeRemained = sizeRemained – ***sizeMax***
15: **end while**
16: S_i.data = $\boldsymbol{D_{Bigdata}}$[(i − 1)×***sizeMax*** + 1: (i − 1)×***sizeMax*** **+** sizeRemained]
17: S_i.dataSize = sizeRemained
18: w_j **= W[j]**
19: $\boldsymbol{D_{RAPW-RLE}}$.append(**RAPW-RLE(**$\boldsymbol{\tau}$, S_i, w_j**)**) /* in Algorithm 1 */

Algorithm 3. RAPW-ReadWM algorithm for reading watermark from a RAPW-BEP data stream.

Input: bounded-error $\boldsymbol{\tau}$, RAPW-LBE data stream $\boldsymbol{D_{RAPW-BEP}}$, original stream $\boldsymbol{D_{Bigdata}}$, maximum size ***sizeMax*** of sub-stream, watermark robustness threshold ***wrThreshold***

Output: Watermark W[]

1: $N = \boldsymbol{D_{RAPW-BEP}}$.dataSize
2: sizeRemained = N
3: $I = k = i = j = 1$
4: countofDelimiter = 0
5: lowerboundData[:] = $\boldsymbol{D_{Bigdata}}$ [:] $- \boldsymbol{\tau}$
6: countofWK[0:*l*MAX] = 0 /* clear counts of all possible $w_{\underline{k}}$ */
7: **while** (sizeRemained > ***sizeMax***)
8: $w_k = (\boldsymbol{D_{RAPW-LBE}}[I].\boldsymbol{d_i}$.staringPoint – lowerboundData[j])/0.1
9: countofWK[w_k]= countofWK[w_k] + 1 /* count occurrence of w_k */
10: **if** ($w_k = w_{delimiter}$)
11: $k = 1$
12: countofDelimiter ++
13: **if** (countofDelimiter = 2)
14: **return** W[] /* confirm the watermark integer list */
15: **end if**
16: **end if**
17: $j = j + \boldsymbol{D_{RAPW-LBE}}[I].\boldsymbol{d_i}$.runLength
18: $i = i + 1$
19: **if** ($j >$ ***sizeMax***)
20: I++ /* jump to new sub-stream as defined in Equation (9) */
21: MaxcntInx = index of maximum count in CountofWK[]
22: Othercnt = total of CountofWK[] except index of MaxcntInx
23: **if** (Othercnt $\leq$ ***wrThreshold***)
24: W[k] = MaxcntInx
25: $k = k + 1$
26: **else**
27: return NIL /* return no watermark and exit*/
28: **end if**
29: sizeRemained = sizeRemained $-$ ***sizeMax***
30: $i = j = 1$ /* first data in new sub-stream */
31: countofWK[0:*l*MAX] = 0 /*clear counts of all possible $w_{\underline{k}}$ */
32: **end if**
33: **end while**
34: /* check the remained sub-stream if meeting the conditions of step 13 to 15 in **while loop,** otherwise return no watermark and exit */

4. Experiments and Evaluations

The main sources of our experimental data were divided into four different kinds of datasets, including Brazil's underwater temperature [21], London's ultraviolet index [22], and the USA's COVID-19 confirmed cases and deaths in New Jersey [23]. Each dataset had the same 500 data values with different attributes and ranges.

4.1. Preliminary Experiments for IoT Data Owners and Consumers

However, we first used the original data sub-stream from Figure 2 to conduct the preliminary testing for the IoT data owners and consumers of this data sub-stream. As shown in Figure 5, it validated the DAPW-RLE sub-streams in different watermark integers (i.e., ranging from 1 to 19 in Equation (14)) within the assigned bounded-error of 1. These RAPW-RLE subs-streams preserved a similar shape to the original sub-stream (i.e., red line) with privacy protection of anonymity for consumers, who subscribed a RAPW-BEP data stream from an owner. Then, this RAPW-RLE data sub-stream could tell the owner's ownership from the inserted watermark integer.

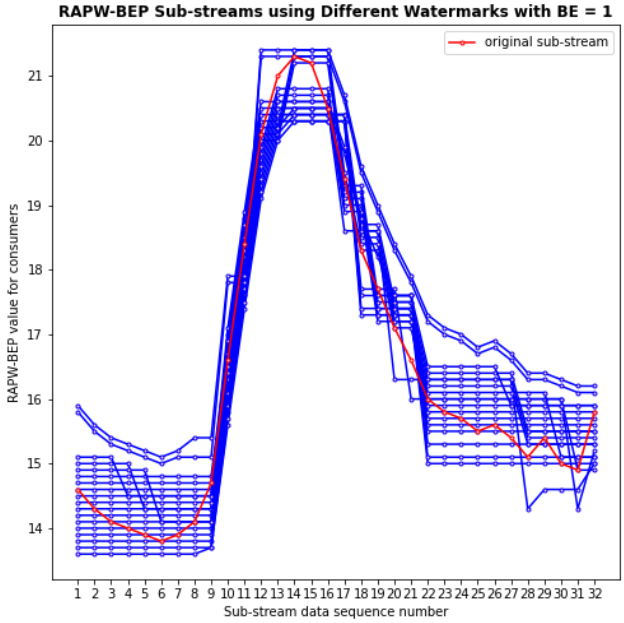

Figure 5. RAPW-BEP sub-streams with different single watermark integers (1 to 19) with $\tau = 1$.

Then, to simply validate if this RAPW-RLE sub-stream with watermarking can still preserve the resource efficiency in lossless compression with different assigned bounded-errors (i.e., BE = 1.0 and 0.5), we used this original data sub-stream again to conduct the preliminary testing for the entropy value (as calculated in Equation (7)) of this IoT data sub-stream using different possible watermark integer. As shown in Equations (14) and (15), the watermark integer for a single sub-stream was ranged from 1 to 19 if BE equaled 1.0 and from 1 to 9 if BE equaled 0.5. In Figure 6, these entropy results indicate that all the entropies of RAPW-BEP sub-streams with different watermark integers were much lower than the original sub-stream in lossless compression for better resource efficiency.

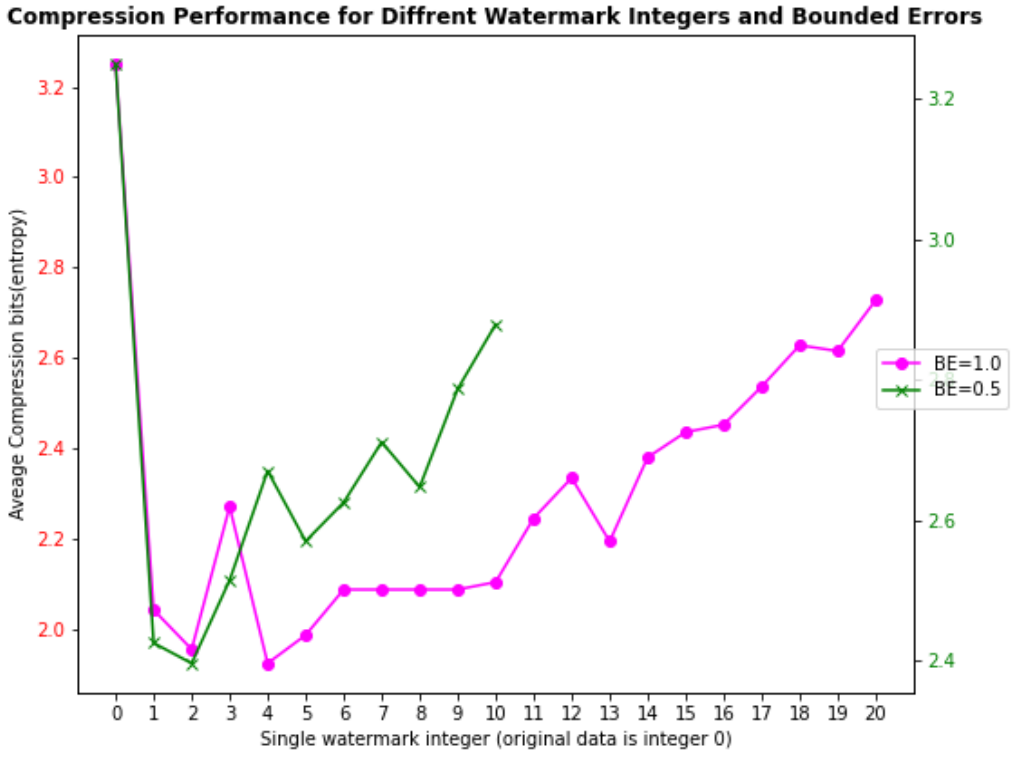

Figure 6. Entropies for a RAPW-RLE-encoding sub-stream via different watermark integers and BE.

4.2. Examine for RAPW-BEP IoT Data Streams with Different Settings

To further demonstrate the feasibility of our proposed RAPW schemes, we used four different kinds of datasets from open data. They were the underwater temperature dataset [21] with slow variation in larger data values, the UV index dataset [22] with slow variation in smaller data values, and the COVID-19 dataset [23] with confirmed case and death numbers with a very large burst in much larger values and smaller values, respectively. In these different-type datasets, we equally picked up 500 sequential data among them as the input for our proposed RAPW scheme, as shown respectively in the RAPW-RLE and RAPW-BEP algorithms of Algorithms 1 and 2, to output the RAPW-BEP data streams for owners and consumers in IoT big data market. Since the long IoT data stream is usually segmented into small-sized sub-stream for the required service response time, the applied sub-stream sizes were all the same 32 in the first experiment for these four different kinds of datasets.

As simply shown in Equation (16) of our proposed RAPW scheme, each RAPW-BEP sub-stream possessed a watermark integer from the IoT data owner's watermark integer string sequentially. The watermark integer 0 was considered a delimiter to represent the end of the watermark integer string. Therefore, the watermark string could be told from the long-running sub-stream in the whole RAPW-BEP data stream. To provide sufficient availability in the watermark integer string for owner's ownership representation, because different bounded errors will have the inherent limitation of a watermark integer range, our applied lengths of watermark integer string in testing were 4, 6, 8, and 10 (not including the delimiter) for these four different kinds of datasets, respectively.

In Figure 7, the RAPW-BEP data stream for Brazil's underwater temperature dataset is shown in the blue line in the figure top, and the green line is the original temperature dataset. The applied bounded-error for this temperature data stream was 1.0. In the lower part of Figure 7, the square line in gray is the repeated watermark integer strings. The watermark integer 0 indicates the delimiter between the same watermark strings. These watermark integers in the watermark integer string should be assigned from the input watermark from the data owner. In our experiments, all the watermark integer strings were randomly generated with different lengths for fair testing. The applied length of the watermark integer string in Figure 7 was 4.

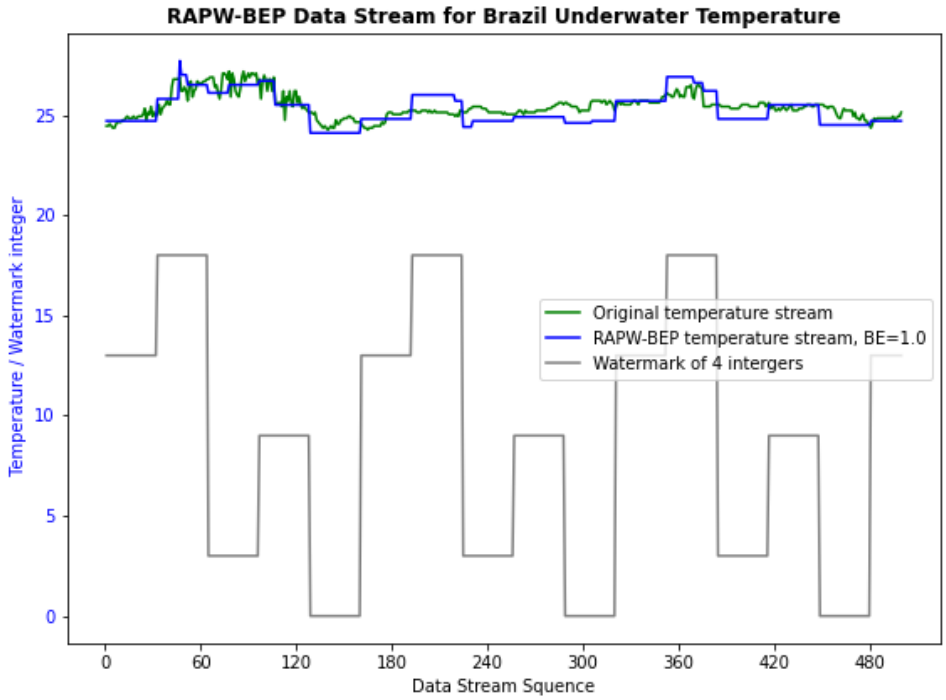

Figure 7. Data sequence of RAPW-BEP temperature data stream with 4 watermark integers.

In Figure 8, the RAPW-BEP data stream for the UV index dataset in London's area is shown in the blue line in the figure top, and the green line is the original temperature dataset. The applied bounded-error for this temperature data stream was 0.5. In the lower part of Figure 8, the gray square line is the repeated watermark integer strings with a length of 6. The RAPW-BEP data stream for the COVID-19 confirmed dataset of New Jersey, USA is shown in the blue line at the bottom of Figure 9. The vague green line at the bottom of the figure is the original confirm dataset. The gray square line is the repeated watermark integer string with a length of 8. In Figure 10, the RAPW-BEP COVID-19 death dataset is shown in the blue line in the figure bottom and the original dataset is the vague green line. At the top of the figure, the gray square line is the repeat watermark integer strings with a length of 10. All the applied confirmed and death datasets possess a large burst of data value with different scales. We believe these four different-type datasets validate the performance of our prosed RAPW scheme comprehensively.

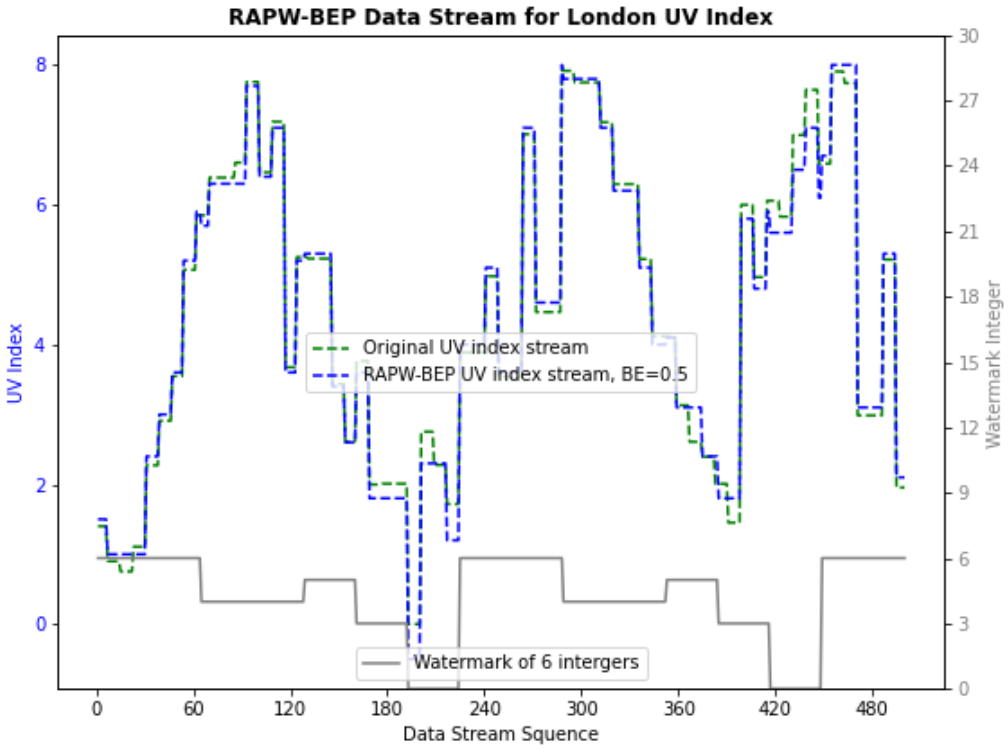

Figure 8. Data sequence of RAPW-BEP temperature data stream with six watermark integers.

4.3. Overall Performance Evaluation with Watermarking Robustness

To fully demonstrate the resource efficiency and watermarking robustness in the proposed RAPW scheme, we applied different settings of bounded-error, watermark length (WL), and sub-stream size (SS) to find out the corresponding compression ratio (as shown in Equation (19)) and robustness ratio (as shown in Equation (21)).

In Equation (19), the improvement of the compression ratio of the RAP-BEP data stream is defined by dividing the compression bits of the RAPW-BEP data stream by the compression bits of the original data stream. The compression bits of the RAPW-BEP data stream can be obtained by multiplying the length of the RAPW-BEP stream (i.e., $n_{RAPW-BEP}$) and RAPW-BEP stream's entropy (i.e., $entropy(D_{RAPW-BEP}(\tau, \omega))$, a similar calculation as in Equation (7)). n is the length of the original data stream, τ is the bounded-error, ω is the applied watermark, and the entropy of the original data stream (i.e., $entropy(D_{sensor}(n))$ was defined in Equation (6).

$$Compression\ Ratio = \frac{RAPW - BEP\ Data\ Stream\ Compression\ Bits}{Original\ Data\ Stream\ Compression\ Bits} = \frac{n_{RAPW-BEP} \times entropy(D_{RAPW-BEP}(\tau, \omega))}{n \times entropy(D_{sensor}(n))} \quad (19)$$

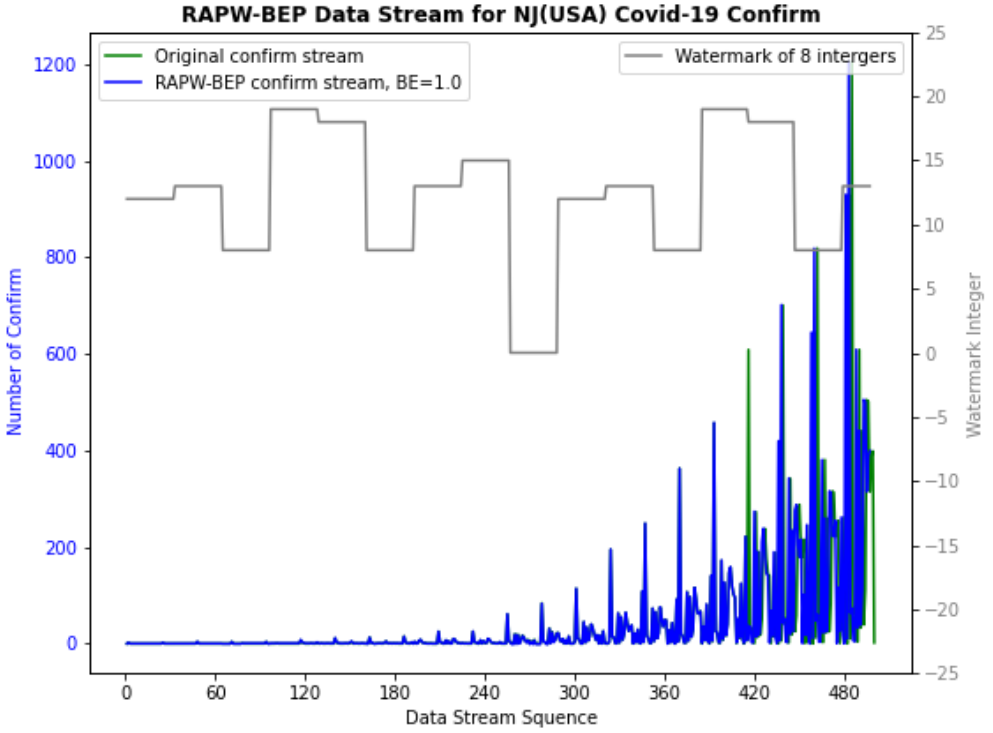

Figure 9. Data sequence of RAPW-BEP COVID-19 confirm data stream witheight watermark integers.

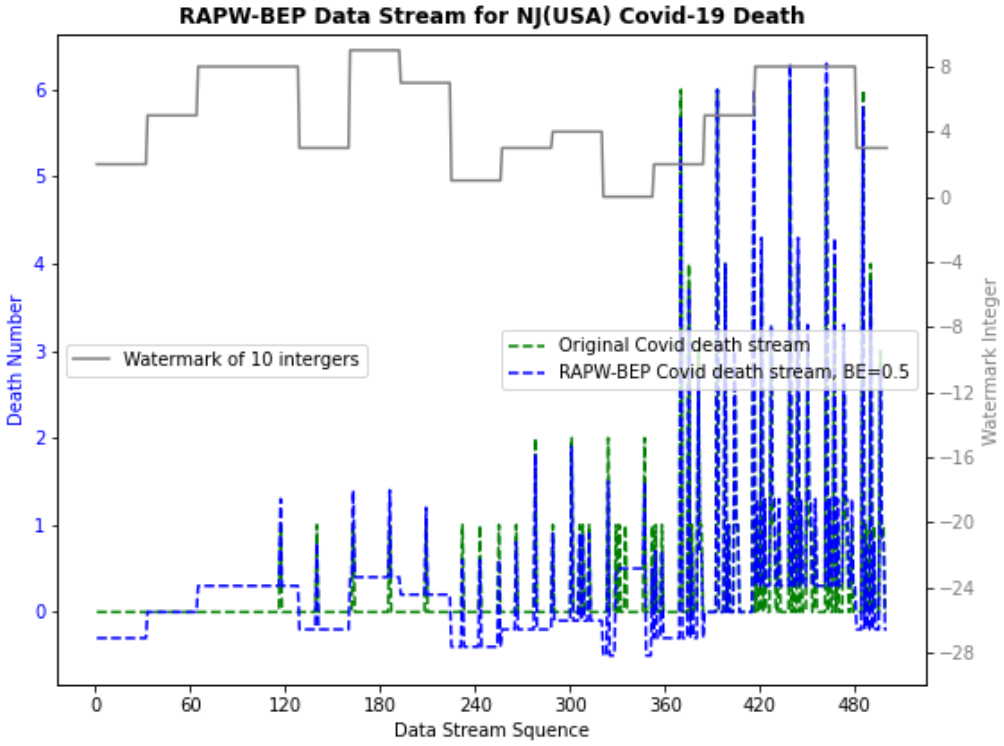

Figure 10. Data sequence of RAPW-BEP COVID-19 death data stream with ten watermark integers.

With Equation (19), we applied different bounded-errors of 1.0 and 0.5, watermark lengths (WL) of 4, 8, and 12, sub-stream size (SS) of 16, 32, and 48 on four different-type datasets to validate the compression ratio performance of our proposed RAPW scheme in resource efficiency in the IoT big data market. The compression performance results for two different bounded-errors are illustrated in Figures 11 and 12, respectively. Then, we found that the high-bounded-error usually had a better compression ratio than the

low-bounded-error for the RAPW-BEP data stream. It is notable that the smaller IoT data values with a lower variation preserved a better compression ratio in the RAPW-BEP data stream, no matter what the different settings of watermark length (WL) and sub-stream size (SS) are assigned in experiments. Thus, the RAPW-BEP data stream of the COVID-19 death dataset had a poor compression ratio since its large-value burst with high variation.

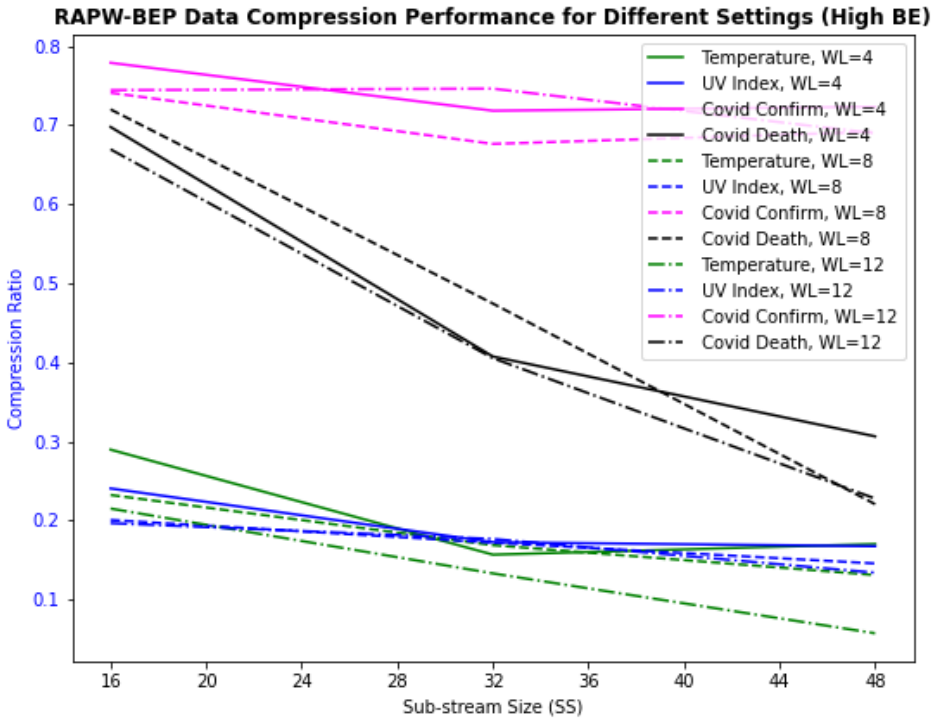

Figure 11. Compression ratio of different RAPW-BEP datasets with high-bounded-error.

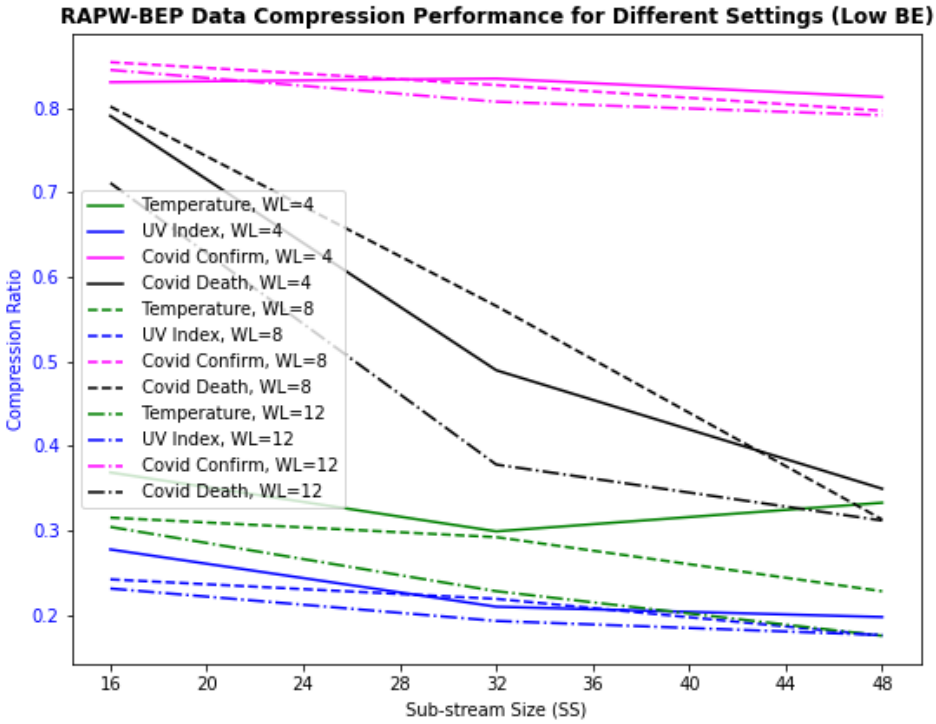

Figure 12. Compression ratio of different RAPW-BEP datasets with low-bounded-error.

Without the loss of generality, in the COVID-19 confirmed and death datasets, all the integer data values were first normalized to have one decimal place and were then applied to bounded-errors of 1.0 and 0.5 for the performance evaluations of proposed RAPW schemes. Thus, the actual bounded-errors in the confirmed and death cases in the COVID-19 dataset were 10 and 5, respectively.

In the proposed RAPW scheme, the integers in the watermark integer string for the data owner were sequentially inserted into the RAPW-BEP sub-streams from the original data streams. All the watermark integer strings with different lengths applied in our experiment on four different-type datasets were generated at random to fairly demonstrate the watermarking performance. All the watermark extraction results are listed in Table 1 to indicate if the watermark could be correctly read from the RAPW-BEP data streams of these four different-type IoT datasets.

Table 1. Extraction results of random watermarks for four different datasets with different settings.

Watermark Length, Sub-Stream Size/Dataset	WL = 4, SS = 16	WL = 8, SS = 16	WL = 12, SS = 16	WL = 4, SS = 32	WL = 8, SS = 32	WL = 12, SS = 32	WL = 4, SS = 48	WL = 8, SS = 48	WL = 12, SS = 48
Temperature (*BE = 1.0*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
UV Index (*BE = 1.0*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
COVID Confirm (*BE = 1.0*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
COVID Death (*BE = 1.0*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
Temperature (*BE = 0.5*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
UV Index (*BE = 0.5*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
COVID Confirm (*BE = 0.5*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.
COVID Death (*BE = 0.5*)	ok	ok	ok	ok	ok	ok	ok	ok	Incomp.

As shown in Table 1, only the results from the setting of "WL = 12 and SS = 48" were not "ok" in all testing datasets. This is because the sizes of the original data stream in our testing datasets were 500. If the watermark string length was 12 and the sub-stream size was 48 in the applied test setting, the size of the original data stream should be larger than $(12 + 1) \times 48$ (i.e., 624) to read a complete watermark string correctly. As shown in the constraint of the data stream size n in Equation (20), the reason why the incomplete watermark was found in the extraction resulted from the setting of watermark string length of 12 and sub-stream size of 48.

$$n \geq (WL + 1) \times SS \tag{20}$$

However, the watermark of RAPW-BEP data streams of anonymous consumers is used to identify the IoT data owner's ownership in owner's RAPW-BEP data streams for consumers' anonymity using different data resolutions (i.e., bounded-errors) with different prices. According to the blockchain system flowchart described in Figure 3 using the proposed RAPW scheme, the owner's original data and the owner's RAPW-BEP data streams with different data resolutions in BE are all stored on the P2P IPFS system with their unique hash values. Though these unique hash values, which can be stored on the Ethereum blockchain via a smart contract, can guarantee their unchangeable content including owners' original data and their BEP data for consumers, the security notions for our proposed approach are still concerned that the anonymous consumers may either accidentally or intentionally tamper or even resell these downloadable BEP contents without authorization. To correctly read out the watermark embedded in the downloaded BEP data stream, our proposed RAPW-ReadWM algorithm, as illustrated in Algorithm 3, must include the owner's original data stream in P2P IPFS, the bounded-error value τ in blockchain, and the consumers' downloaded BEP data stream which can be tampered.

Considering the above-mentioned security notions about the blind watermark removal and tampering attacks (i.e., with perfect knowledge of the watermark scheme) in the threat models [24] for our proposed watermark scheme, we further conducted the experiment to validate the robustness of watermarking in the proposed RAPW scheme. Using the given tempered data ratios of 1%, 2%, 10%, and 20%, we then randomly selected the data values in the data stream to change their value within the given BE setting. This experiment on

the four different-type testing datasets used all the same settings as in Table 1. Then, the RAPW-ReadWM algorithm in Algorithm 3 was used to read the inserted watermark out from all tampered data streams with different tampered ratios of 1%, 2%, 10%, and 20%. All the testing with different tampered ratios for the validation of watermark correctness were repeated 15 times to find out the successful ratio on average for RAPW watermarking robustness.

$$Robustness\ ratio = \frac{\#\ of\ successful\ watermark\ extractions}{\#\ of\ inserted\ watermarms} = 1 - \frac{\#\ of\ mismatched\ and\ imcomplete\ watermarks}{\#\ of\ inserted\ watermarks} \quad (21)$$

The so-called robustness ratio is defined in Equation (21). The mismatched watermarks and incomplete watermarks (as shown in Table 1) were all culled to demonstrate the robustness ratio in strictness. The experimental result of the robustness ratio for RAPW watermarking is shown in Table 2. The COVID-19 confirmed dataset with the largest burst of data values had the worse robustness, since it possessed the smallest number of run-length coding to protect the inserted watermark integer from tampering.

Table 2. Robustness ratio of RAPW for four different datasets from different tampered ratios.

Tampered Ratio/Dataset	1%	2%	10%	20%
Temperature	83%	77%	56%	30%
UV index	83%	78%	50%	28%
COVID confirmed cases	63%	58%	32%	21%
COVID death	83%	76%	41%	14%

5. Conclusions and Future Work

The concept of bounded-error-pruned (BEP) IoT content was used in this paper to propose new resource-efficient anonymity protection with watermarking (RAPW) for the IoT big data market via an Ethereum blockchain smart contract. In this decentralized IoT big data market, the owner's IoT big data and their different RAPW-BEP IoT data for anonymous consumers were securely stored on the P2P IPFS file system with immutability and availability.

Through the experiments on the four different-type datasets including large data values with small variations, small data values with small variations, large bursts in large data values, and large bursts in small data values, the proposed watermarking scheme in RAPW-BEP IoT data can further provide data ownership protection for owners with robustness without degrading the inherent resource efficiency in BEP IoT big data. Thus, we believe that our proposed RAPW scheme can encourage more IoT big data owners including public sectors to provide consumers valuable data with anonymity for further fostering more diversified IoT applications.

In the future, we hope to validate and enhance our RAPW performance on not only the more diversified and complicated IoT datasets, such as visual and audible IoT datasets, but also by comparing and consolidating with other IoT watermark techniques [17] in cost efficiency, for better robustness in watermark protection from the data tampering from malicious users in IoT big data markets.

Author Contributions: Conceptualization, C.-H.W.; methodology, C.-H.W.; software, C.-H.W.; validation, C.-H.W.; formal analysis, C.-H.W.; investigation, C.-H.W. and C.-H.H.; data curation, C.-H.W. and C.-H.H.; writing—original draft preparation, C.-H.W. and C.-H.H.; writing—review and editing, C.-H.W.; visualization, C.-H.W. and C.-H.H.; funding acquisition, C.-H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MOST Taiwan, grant number 110-2221-E-130-001 and the APC was funded by 110-2221-E-130-001.

Acknowledgments: The authors gratefully thank the financial support from the Ministry of Science and Technology, Taiwan (MOST 110-2221-E-130-001).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garrido, G.M.; Sedlmeir, J.; Uludağ, Ö.; Alaoui, I.S.; Luckow, A.; Matthes, F. Revealing the landscape of privacy-enhancing technologies in the context of data markets for the IoT: A systematic literature review. *J. Netw. Comput. Appl.* **2022**, *207*, 103465. [CrossRef]
2. Manzoor, A.; Braeken, A.; Kanhere, S.S.; Ylianttila, M.; Liyanage, M. Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain. *J. Netw. Comput. Appl.* **2021**, *176*, 102917. [CrossRef]
3. General Data Protection Regulation (GDPR). Available online: https://gdpr-info.eu (accessed on 21 December 2021).
4. Chang, R.-I.; Wei, L.-C.; Wang, C.-H.; Tseng, Y.-K. Blockchain for bounded-error-pruned content protection. *ICT Express* **2021**, *7*, 295–299. [CrossRef]
5. Furon, T. A Constructive and Unifying Framework for Zero-Bit Watermarking. *IEEE Trans. Inf. Forensics Secur.* **2007**, *2*, 149–163. [CrossRef]
6. Gourrame, K.; Douzi, H.; Harba, R.; Riad, R.; Ros, F.; Amar, M.; Elhajji, M. A zero-bit Fourier image watermarking for print-cam process. *Multimed. Tools Appl.* **2019**, *78*, 2621–2638. [CrossRef]
7. Roček, A.; Javorník, M.; Slavíček, K.; Dostál, O. Zero Watermarking: Critical Analysis of Its Role in Current Medical Imaging. *J. Digit. Imaging* **2021**, *34*, 204–211. [CrossRef] [PubMed]
8. Oliva, G.A.; Hassan, A.E.; Jiang, Z.M. An exploratory study of smart contracts in the Ethereum blockchain platform. *Empir. Softw. Eng.* **2020**, *25*, 1864–1904. [CrossRef]
9. Azar, J.; Makhoul, A.; Barhamgi, M.; Couturier, R. An energy efficient IoT data compression approach for edge machine learning. *Future Gener. Comput. Syst.* **2019**, *96*, 168–175. [CrossRef]
10. Marcelloni, F.; Vecchio, M. Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization. *Inf. Sci.* **2010**, *180*, 1924–1941. [CrossRef]
11. Liu, Y.; Di, S.; Zhao, K.; Jin, S.; Wang, C.; Chard, K.; Tao, D.; Foster, I.; Cappello, F. Optimizing Error-Bounded Lossy Compression for Scientific Data with Diverse Constraints. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 4440–4457. [CrossRef]
12. Chang, R.-I.; Chu, Y.-H.; Wei, L.-C.; Wang, C.-H. Bounded-Error-Pruned Sensor Data Compression for Energy-Efficient IoT of Environmental Intelligence. *Appl. Sci.* **2020**, *10*, 6512. [CrossRef]
13. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
14. Benet, J. Ipfs-content addressed, versioned, p2p file system. *arXiv* **2014**, arXiv:1407.3561.
15. Ali, M.S.; Dolui, K.; Antonelli, F. IoT data privacy via blockchains and IPFS. In Proceedings of the Seventh International Conference on the Internet of Things, Linz, Austria, 22–25 October 2017; pp. 1–7.
16. Radanliev, P.; De Roure, D.; Walton, R.; Van Kleek, M.; Montalvo, R.M.; Maddox, L.; Santos, O.; Burnap, P.; Anthi, E. Artificial intelligence and machine learning in dynamic cyber risk analytics at the edge. *SN Appl. Sci.* **2020**, *2*, 1773. [CrossRef]
17. Wazirali, R.; Ahmad, R.; Al-Amayreh, A.; Al-Madi, M.; Khalifeh, A. Secure Watermarking Schemes and Their Approaches in the IoT Technology: An Overview. *Electronics* **2021**, *10*, 1744. [CrossRef]
18. Buterin, V. A Next-generation Smart Contract and Decentralized Application Platform. *Ethereum White Pap.* **2017**, *3*, 1–2.
19. Li, Z.N.; Drew, M.S.; Liu, J. *Fundamentals of Multimedia*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2004.
20. Chang, R.-I.; Tsai, J.-H.; Wang, C.-H. Edge Computing of Online Bounded-Error Query for Energy-Efficient IoT Sensors. *Sensors* **2022**, *22*, 4799. [CrossRef]
21. An Underwater Temperature Dataset from Coastal Islands in Brazil. Available online: https://www.kaggle.com/datasets/shivamb/underwater-surface-temperature-dataset (accessed on 29 July 2022).
22. UV Index Data Collected Local to West London (Heathrow and Northolt). Available online: https://www.kaggle.com/datasets/t5ra190/uv-index-dataset-local-to-west-london (accessed on 29 July 2022).
23. COVID-19 Data Repository. Available online: https://github.com/CSSEGISandData/COVID-19 (accessed on 29 July 2022).
24. Li, Q.; Memon, N. Security models of digital watermarking. In *International Workshop on Multimedia Content Analysis and Mining*; Springer: Berlin/Heidelberg, Germany, 2007.