Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information). 10/1 (2017), 1-10
DOI: http://dx.doi.org/10.21609/jiki.v10i1.440

DE-IDENTIFICATION TECHNIQUE FOR IOT WIRELESS SENSOR NETWORK PRIVACY PROTECTION

Yennun Huang[1], Szu-Chuang Li[1], Bo-Chen Tai[1], Chieh-Ming Chang[1], Dmitrii I. Kaplun[2], and Denis N. Butusov[2]
1CITI, Academia Sinica, 128 Academia Road, Section 2, Nankang,, Taipei City, 115, Taiwan
2Saint Petersburg Electrotechnical University "LETI", ul. Professora Popova 5, 197376 Saint Petersburg, Russian Federation

E-mail: yennunhuang@citi.sinica.edu.tw

Abstract

As the IoT ecosystem becoming more and more mature, hardware and software vendors are trying create new value by connecting all kinds of devices together via IoT. IoT devices are usually equipped with sensors to collect data, and the data collected are transmitted over the air via different kinds of wireless connection. To extract the value of the data collected, the data owner may choose to seek for third-party help on data analysis, or even of the data to the public for more insight. In this scenario it is important to protect the released data from privacy leakage. Here we propose that differential privacy, as a de identification technique, can be a useful approach to add privacy protection to the data released, as well as to prevent the collected from intercepted and decoded during over-the-air transmission. A way to increase the accuracy of the count queries performed on the edge cases in a synthetic database is also presented in this research.

Keywords: *differential privacy, internet of things, sensor network*

Abstrak

Sebagai ekosistem IOT menjadi lebih dan lebih dewasa, vendor hardware dan software berusaha menciptakan nilai baru dengan menghubungkan semua jenis perangkat bersama melalui IOT. Perangkat IOT biasanya dilengkapi dengan sensor untuk mengumpulkan data, dan data yang dikumpulkan ditransmisikan melalui udara melalui berbagai jenis koneksi nirkabel. Untuk mengekstrak nilai data yang dikumpulkan, pemilik data dapat memilih untuk meminta bantuan dari pihak ketiga dalam analisis data, atau bahkan data kepada publik untuk wawasan yang lebih dalam. Dalam skenario ini penting untuk melindungi data yang dirilis dari kebocoran privasi. Di sini kami mengusulkan bahwa privasi diferensial, sebagai teknik identifikasi de, dapat menjadi pendekatan yang berguna untuk menambah perlindungan privasi data yang dirilis, serta untuk mencegah diambil dan diterjemahkan selama transmisi over-the-air. Sebuah cara untuk meningkatkan akurasi query count dilakukan pada kasus tepi dalam database sintetis juga disajikan dalam penelitian ini.

Kata Kunci: *privasi differensial, internet of things, jaringan sensor*

1. Introduction

As the IoT ecosystem becomes more and more mature in recent years, hardware and software vendors are trying to create new value by connecting all kinds of devices together via IoT. One of the primary functions of an IoT device is to collect and transfer data using equipped sensors. Rapid and enormous data collection has been happening in the past years on PC and mobile phones. According to IBM during the last few years 2.5 billion gigabytes of high-velocity data, such as social media posts, information gathered from sensors and medical devices, videos and transaction records, are created in a variety of forms every day, and the rise of the IoT devices in numbers will cause the quantity of data collected each day to skyrocket. Gartner[1] predicts that in 2016 there're already 6.4 billion IoT devices, and the number will be tripled in 2020, making it 20.8 billion.

IoT devices possess very different qualities than a PC or mobile phone. First, they're often deployed in large number: in the future we might have several wearable devices per person, as well as multiple IoT-enabled electronics in a household. Second, a lot of IoT devices will be deployed outdoors, and those devices will be vulnerable to physical hacking, and the transmitted data

[1] http://www.gartner.com/newsroom/id/3165317, retrieved on Jan. 26th, 2017

might be intercepted, causing every kind of possibility of privacy leakage. Last, IoT devices usually possess very limited storage and computing resource, making it difficult to use advanced encryption schemes to protect data storage and transmission.

De-identification techniques can be an effective alternative to deal with privacy preserving data transmission and analysis in for IoT. Existing de-identification methods such as K-anonymity and its derivatives, and differential privacy-compliant mechanisms consumes relatively small resource while providing data privacy. In this paper we'll first describe a field test we've done at a local theme park, utilizing a custom-built Bluetooth network and proximity tags to collect spatio-temporal data of the visitors, and we'll discuss how we can remove the sensitive attributes from the data while preserving its statistical utility, so that we can release the data to a third-party for further analysis without revealing privacy information. After coping with the problem of privacy preserved IoT data release, we'll take a brief look at a current option to propose how we can use de-identification techniques to protect data transmission.

2. Methods

Collection of Spatio-Temporal Data from a Custom Bluetooth Sensor Network

Ways to collect spatio-temporal data

With the emergence of wearable devices and sensor technology, there have been plenty attempts to collect and analyze spatio-temporal data. The most common used technologies to retrieve positional information are still GPS and Wifi [1-4]. Recently Bluetooth has become a viable choice to provide positioning service, especially in an indoor scenario. Typically the Bluetooth beacons are configured to send out simple ID information. When installed its physical location will be recorded to a database on a central server or a small local database that's attached to an mobile APP. Whenever a mobile device gets near the Bluetooth beacon and receives the ID information broadcasted by the Bluetooth beacon, it will match the ID information against the data stored in a server or local database on the mobile APP and react accordingly. Recently researchers have been trying to get more precise positional information out of Bluetooth beacons by taking Bluetooth signal strength into account and/or combine information from multiple beacons [5]. Another approach, though, is to get positional information via "crowd sensing". Jamil et. Al. [3] had an attempt to combine mobile phones with Bluetooth proximity tags to rebuild the traces of visitors.

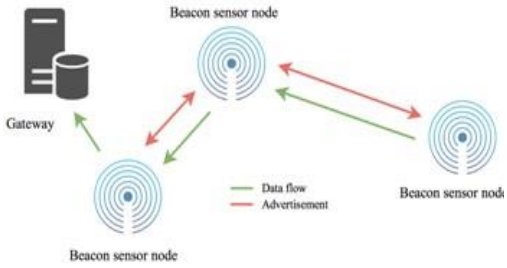

Figure 1. Relaying data through a series of Bluetooth beacons

A custom solution to collect data in a wireless Internet-less environment

As mentioned, the most common usage of Bluetooth beacons is to use them as broadcasting stations. But since Bluetooth specification actually allows a beacon to work in scan and broadcasting mode, it is possible to relay limited information between Bluetooth beacons, while scanning for Bluetooth proximity tags nearby back and forth. This way the beacons can collect the ID information sent by Bluetooth proximity tags and relay them through a series of beacons. At the end of the beacon chain we can setup a PC as a Bluetooth network-to-Internet gateway to relay collected information to a remote cloud server for data storage and analysis, as in Figure 1.

The "Bluetooth Gateway" is a PC bor Server Jonnected to the Internet with a Bluetooth Interface, and each Bluetooth beacon should be placed within the broadcasting range of the next and previous Bluetooth beacons. The Bluetooth beacons are programmed to carry custom payload, enabling them to do two-way communication in the following fashion:

Upstream communication

Figure 2. Custom made Bluetooth beacon

Figure 3. Bluetooth bracelet from Xiaomi Technology

In the connected Bluetooth network it's possible for a beacon to send data to the gateway PC and even to the Internet when needed. The use of custom payload enables arbitrary data to be relaed all the way to the Bluetooth gateway for further processing. The data are Bluetooth mostly device IDs, but it's possible to send control codes too when needed.

Downstream communication
Information regarding all Bluetooth beacons was aggregated at the Bluetooth gateway, making it possible to send control codes downstream to a particular beacon. For example, the gateway PC can send a command to change the scan interval to a particular Bluetooth beacon to change its behavior. It is also possible to send application related information such as a short message or a URL pointer to all the Bluetooth or mobile phones near a particular beacon.

Power efficiency of Bluetooth beacons
To enable easy deployment and allow Bluetooth beacons to run on batteries for extended period of time, scanning interval of the custom-built Bluetooth beacons were configured to rest for 15 seconds after 5 seconds of scanning and Broadcasting. Coupled with the clocked switch which only turns on Bluetooth beacons during the work hours, a Bluetooth beacon can run for 72 days without batteries changed with 2x 3000mAh batteries installed. Please note that the two-way network is not suitable for real time communication. The beacons are configured to scan peridically. Buffering and confirmation mechanism has been designed very carefully to ensure the reliability of data transmission, and the time required for the packets to travel to the destination is long and may vary. In our experiments the transfer time can be as long as 1 minute when the beacon chain is long.

In past researches Internet connections are required to send the collected positional infomation to a remote server. For example, if we want to collect spatio-temporal information of visitors in a theme park for optimizing the visiting experience, the theme park will have to make visitors install mobile APPs and configure properly and provide wireless Internet access for them if they do not have it themselves. It could be expensive and uh-realistic for a theme park do create those infrastructures or to expect every visitor to have an Internet connection subscription. By using a custom Bluetooth beacon network described here we'll give proximity tags to visitors (Bluetooth bracelets or stickers), and setup beacons along the popular paths. As in Figure 1 the beacons can then repay detected ID information all the way to the Internet. This is made possible by the utilization of the CC2541 SoC's programmable chip from TI, which is used to create a custom protocol to relay information through a series of Bluetooth beacons. A local theme park called "Little Ding Dong science theme park" agreed to let the research team setup more than 50 beacons around the theme park. The devices we used to setup this experiment includes:

Custom-made Bluetooth beacons
Inside the beacon container Three are four somponents: (1) A programmable SoC from Texas Instruments with 8051 ALU and integrated Bluetooth functions, (2) An antena, (3) A waterproof case for reliable operation indoor/outdoor, (4) A pair of batteries that allows the beacon to work for several weeks when fully charged

Utilizing the SoC's programmability, we were able to implement some of the key features of the system: (1) Change signal scanning / transmitting interval to increase power efficiency: to increase power efficiency, the interval of scanning time of Bluetooth beacons can be tuned. Extensive experiments were performed for us to learn about the optimal parameters that balance energy and data transferring efficiency. Based on the experiment results we configure the beacons to scan or broadcast for 5 seconds and sleep for 15 seconds. The beacons will also be configured to

Figure 4. Physical placement of Bluetooth beacons

run for 8 hours a day. A beacon equipped with 2 3000mAh batteries can run for 72 days nonstop using this setting. This enables fast deployment and easy maintenance for the Bluetooth beacon Networks, (2) Enabling two-way communication: the beacons are programmed to relay "upstream" and "downstream" data. For instance, identification information of Bluetooth bracelets collected by the beacons will be sent "upstream" to the gateway PC (described later), and will be relay to cloud server thereafter. The gateway PC can send commands "downstream" to a particular beacon through a predefined path. Please note that, to accommodate the energy efficiency arrangements, the two-way communication will not be real-time and will inevitably introduce latency in data transmission.

Sending out identification info: Bluetooth bracelet / proximity tag to send
Bluetooth bracelets from Xiaomi technology are affordable and serve the purpose well. Around 50 units were given to the visitors when they enter the theme park, and the bracelets were returned when they leave the theme park in exchange for coupons that offer a discount when they visit the theme park next time.

Gateway PC with Internet connection and Bluetooth connectivity
There'll be a PC with Bluetooth connectivity and Internet connection at the end of the Bluetooth beacon chain. It will act as a gateway to enable the Bluetooth network to exchange information with the Internet.

Remote cloud server
To analyze the data collected effectively, a remote cloud server with adequate processing power and storage will serve as a storage and data analysis platform. The web server will provide HTTP REST-based API to process data storage requests and attraction recommendation information to users. Route prediction algorithm will also be implemented on the web server.

Setting up the beacons
More than 50 beacons were setup in the theme park to collect spatio-temporal data of the visitors. Since we want to deploy as few beacons as possible, the beacons were tested and it is confirmed that their range of transmission is 15-20 meters. A person will be detected by nearby beacons, and since we're not utilizing signal strength data at this time, placing beacons farther apart will help to reduce redundant detection of visitors from the same beacons. Also since the beacons are placed mostly outdoor, it is important that there're clear path between beacons for Bluetooth signal to be transmitted reliably (no walls present to reflect the signals). In Figure 4, it is shown that the beacons often have to be placed higher above the ground to ensure that there're clear paths between the beacons.

It is worth noting that there're Bluetooth beacons on the market that can run for years on battery, but this is not the case in our study. The custom-built beacons do not just sending out ID information, instead they keeps switching Petween scanning and broadcasting mode, and have to buffer data before relaying them to the other beacons. By carefully tuning the switching interval they still manage to last 8 to 9 weeks before the batteries have to be replaced.

3. Results and Analysis

Following the BLE specification a Bluetooth packet can only be 32-byte in length, and we have to design the transmission data format around this restriction. To ease power consumption, the beacons will detect at most 28 visitors' proximity tag at each round of scanning, and the data will be squeezed into a single packet and transmitted to the next beacon in line. As illustrated in Fig 1. the data will be transmitted along the chain of Bluetooth beacons, all the way to the gateway and eventually to a remote server in the cloud. A MySQL server is installed on the cloud server to store the collected data. We setup a data schema to store such data as n Table 1.

TABLE 1
COLLECTED DATA ATTRIBUTES

Data	Note
Beacon ID	Which beacon detected this bracelet
Bracelet ID	Which bracelet was detected
Timestamp	The time that this data is written to database

From this data we can perform some analysis on the users' visiting behavior. For example, we can reconstruct the route of a particular visitor using the data (Figure 5.), or draw a histogram to show which attractions in the theme park is most visited.

More analysis can be performed on the raw data to gain more insight regarding how the visitors visit the theme park. However, sometimes the data collector doesn't necessarily have the ability do make the most out of the data, hence the need to share those data with a third-party or even release it to the public for further analysis. In this case adequate privacy must be ensured, or the release of such data can violate privacy regulations. We'll discuss how we can protect raw data before release in the following paragraph.

Figure 5. Reconstruction of route for a particular visitor

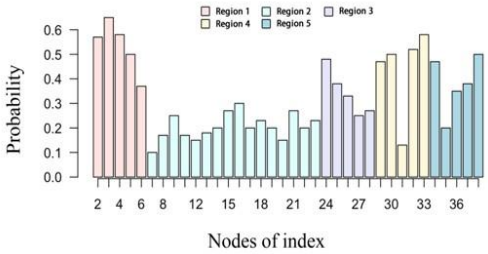

Figure 6. Reconstruction of route for a particular visitor

Ensuring Privacy When Releasing Data to a Third Party or the General Public

It is expected the number of IoT devices will grow rapidly in the coming years. IoT devices not only possess processing power and storage capability, but are also equipped with sensors and actuators. Massive amount of data will be collected by the sensors, and then transferred and stored. Eventually they have to be analyzed to generated value. To ensure privacy of released data, there have been some developed methodology trying to achieve this goal, and those techniques are often labeled as "data de-identification". The more mentioned ones includes K-anonymity [6] and its derivatives[7,8], differential privacy [16], and other attempts from statistical discipline [9]. Due to its deployment by major companies such as Apple[2] and Google, here we'll discuses differential privacy as a potential solution to ensure privacy on IoT data release. It is worth noting that all kinds of data de-indentification techniques so far have to face the problem of privacy-utility tradeoff. The more a data set is processed extensively to hide all the sensitive information, the more decrease in data utility can be expected.

Differential Privacy

Differential Privacy is first proposed by [16], with a provable definition of privacy. The idea is that when one perform a query on a data set (e.g. count number of the entries that fits a set of criterions), the result will be randomized so that the result would not be significantly different whether a particular record presents in the data set or not. The most widely known definition is as below:

Definition 1 [16]. *A randomized κ function gives ε-differential privacy if for all data sets D1 and D2 differing on a t most one element, and all $S \subseteq Range(\kappa)$,*

$$\mathbf{Pr}[\boldsymbol{k}(\boldsymbol{D_1}) \in \boldsymbol{S}] \leq \mathbf{exp}(\boldsymbol{\epsilon})\, \boldsymbol{x}\ \mathbf{Pr}[\boldsymbol{k}(\boldsymbol{D_1}) \in \boldsymbol{S}] \ldots \quad (1)$$

The probability is taken is over the coin tosses of k.

The single record that is different in D1 and D2, can cause privacy leak if the value is vastly different from the other values in the data set. For example, if there's a millionaire in the area, by

[2] Andy Greenberg, Apple's 'Differential Privacy' is about collecting your data, butyou're *your* data, https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/, retrived on Feb. 6th, 2017.

looking at the average income of a data set it could be easy to tell if this person's income is present in the data set or not. So when we decide how much "noise" we want to add to the query result we must take this into account.

Definition 2 [16]. *For f: $D \rightarrow R^k$, the sensitivity of is*

$$\Delta \boldsymbol{f} = \max_{\boldsymbol{D1},\boldsymbol{D2}} \|\boldsymbol{f}(\boldsymbol{D_1}) - \boldsymbol{f}(\boldsymbol{D_2})\|_1 \ldots \quad (2)$$

By withdrawing each record from the data set and calculate the query result on the remaining data entries, we can identify the maximum possible difference the absence of an data entry with extreme value can produce, and take it into account when we decide how much "noise" we should add to a query result to ensure differential privacy. There are some "randomized functions" that fits this definition, but the most commonly used one is Laplace mechanism.

Theorem 1. *For f: $D \rightarrow R^k$, the mechanism K_f that adds independently generated noise with distribution Lap(Δf /ϵ) to each of the j output terms enjoys ϵ–differential privacy [16].*

According to theorem 1, on query function f the privacy mechanism K responds with equation(3).

$$\boldsymbol{f}(\boldsymbol{X}) + (\boldsymbol{Lap}(\Delta \boldsymbol{f}/\boldsymbol{\epsilon}))^{\boldsymbol{k}} \quad (3)$$

will make the query results returned satisfy ϵ–differential privacy.

By adding "noise" to query results, we hope to prevent an advisory from identifying a person by conducting similar queries on a data set. However it is worth noting that by making the same query over and over again the advisory may still learn the real value of a query overtime, so differential privacy it is still needed to limit the query number of a particular person. This is often referred to as "privacy budget." Also one can always choose a larger ϵ to make the noise smaller, but this will result in higher disclosure risk.

Differential Privacy-Compliant Synthetic Database

As we are trying to deal with the problem of data release to a third party, the query-based version of differential privacy does not really suit our needs. [16] also addressed the issue of "non-interactive differential privacy" and proposed that a synthetic dataset can be generated from the results of a series of counting queries performed on the source data. Essentially, one can first identify all the possible value combinations of the attributes in a data set, and count the occurrence of each instance. According to Definition 2. the sensitivity of count queries is a fixed "1", as when we remove or add a data entry to a data set, the result of counting query will be at most "1". This makes the calculation of sensitivity extremely simple. There are several ways suggested by [16] to generate synthetic data set from counting query results, and below we will describe two of the three approaches she recommended.

The first approach is to simply add Laplace noise to each of these counting results, and rebuild a data set from those counting information. Since

TABLE 2
THE ORIGINAL DATA SET

Age	Height	Weight	Income	TRV	HTN	DGF
64	159	66	39	11	0	0
53	178	78	39	13	0	0
53	168	61	35	9	0	0
57	172	78	50	12	0	1
64	173	53	35	8	0	0

TABLE 3
THE SYNTHETIC DATA SET

Age	Height	Weight	Income	TRV	HTN	DGF
66.5	165.5	71.5	27.83	4.5	0	1
47.5	171.5	77.5	78.44	36.5	0	0
55.5	168.5	79.5	54.34	51.5	0	0
54.5	142.5	87.5	90.49	17.5	1	1
61.5	169.5	96.5	91.7	34.5	0	1

the synthetic data set is built from a series of counting query results that is protected by differential privacy, the data set should preserve privacy well. However, the number of count queries that needs to be done using approach can be excessive large, thus if the source data set is large with multiple attributes and value variation, the calculation time needed will be excessively large. Also although the noise added to each cell of this "contingency table" is relatively small, any query for a marginal (aggregate counting queries that fits certain conditions) can be too large for the result to be useful.

The second approach proposed by [16] is to produce some subset of the "contingency table", which are called "marginal tables", and to connect them together via probabilistic inference mechanism. Some of the attempts of this approach are PrivBayes [10] and DPTable [11], and in this research we use the latter and improve it with ways to improve accuracy without sacrificing privacy, which we'll describe later. Here we'll first describe a the steps involved in DPTable to generate a synthetic data set that can preserve most of the statistical properties of the original data set [12]: (1) Calculating the pair wise mutual information value between attributes. When mutual information value exceeds a certain preset threshold the relationship between the attributes will be preserved in the following process. Noise will be added to the mutual information calculated. (2) Based on step 1. Dependency graph will be constructed. The graph will also be "triangulated" for further processing. (3) The dependency graph will be converted to a junction tree, upon which marginal tables will be built. (4) Noise will be added to the marginal in the marginal tables. (5) The marginal tables as a whole will act as a joint distribution from which new dataset can be synthesized. (6) The data user will then be able to sample arbitrary number of data rows from the joint distribution.

To test DPTable, we made an artificial data set with columns age, height, weight, income, travel, high blood pressure (binary flag) and diabetes (binary flag) attributes. The data set has 100,000 rows. For reference the first 5 rows of the original data set is as Table 2, and the first 5 rows of the synthetic data set is as Table 3.

Please note that Table 3 was not "converted" from Table 2. As described in the step-by-step of DPTable, the DPTable mechanism uses the information in Table 2. To build a joint distribution, and then samples data from the joint distribution to build Table 3. To compare the statistical properties of the original data set and the synthetic data set, we calculate the average and standard deviation of each attributes in the table for a rough comparison. Please note that the attribute "HTN" and "DGF" are binary attributes, so in the "average" column we show the counts of positive ("1") value in those attributes.

In Table 4 we can see that the difference between the average value of INCOME and TRV is larger at around 8% and 31% respectively. For other attributes the difference in average value is quite small. For the binary attribute counts, the synthetic data produces 26% error for HTN and 4% error for DGF respectively. Overall the average values of different attributes are preserved quite well in the synthetic data set. For standard deviation the error for most attributes are significantly higher. Please note, though, this synthetic data set is generated using a small ϵ parameter at 0.01, which means that privacy is very well-protected. If one wishes to favor precision over privacy protection, he or she can always select a larger ϵ.

Use "K-aggregation" to improve the privacy-utility tradeoff in differential privacy compliant synthetic data

Besides tuning the ϵ parameter, researchers are actually trying to find ways to improve the techniques to improve privacy without sacrificing utility or vice versa. For example [13] states that by pre-

TABLE 4
A COMPARISON BETWEEN ORIGINAL AND SYNTHETIC DATA

	Average		Standard Deviation	
	Original	Synthetic	Original	Synthetic
Age	53.32771	52.99481	7.804086	7.670179
Height	168.8197	165.9513	7.972777	13.30099
Weight	77.05943	77.77396	7.718009	10.98742
Income	71.91315	78.20495	25.18623	31.03803
TRV	26.43248	34.69463	11.68337	16.89114
HTN(+)	22187		28072	
DGF(+)	28536		29900	

VALUE	1	2	3	4	5	6	7	8	9	10	11	12
COUNT	1	2	2	4	5	8	10	9	7	3	1	1

↓ ↓ ↓ ↓

VALUE	1~3	4	5	6	7	8	9	10~12
COUNT	5	4	5	8	10	9	7	5

Figure 7. Procudure of K-aggregtion when k = 4

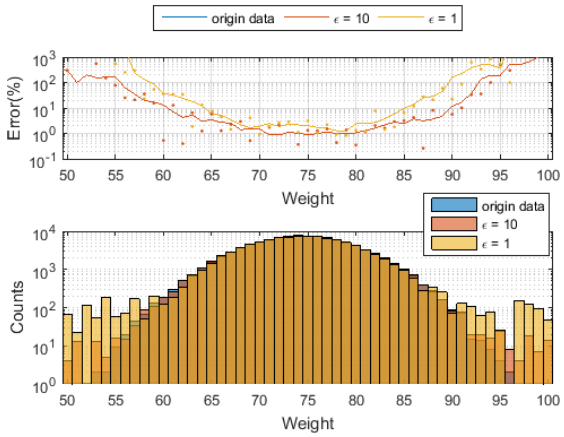

Figure 8. Error % is larger at the edge of a normally distributed dataset due to fewer data counts.

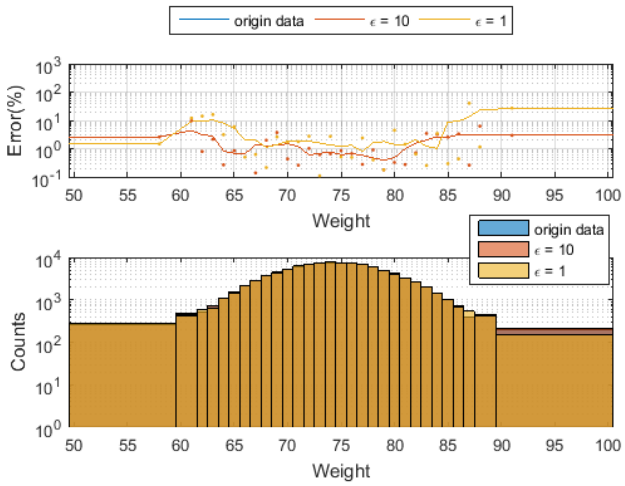

Figure 9. Error percentage variation for the attribute "Weight" with K-aggregation, k=200.

process the dataset using k-anonymity, the amount of noise addition can be reduced to achieve the same privacy in differential privacy, improving accuracy without sacrificing privacy. We examined the procedure and results of DPTable carefully, and here we'll discuss about the ways to improve data utility -- K-aggregation [12].

For data that is normally distributed, there are always fewer counts in extreme cases. For example, people that are extremely tall or short are tend to be small in number and people with more average height would be large in number in a normal distributed data set. As specified previously to apply differential privacy to a tabular data set we first convert to a series of "marginal tables", and then start to add fixed amount of noise to each of the counting query results, and this make it obvious that, proportionally, the marginal (count queries) with fewer count will be influenced by the noise added much more than the marginal with larger counts. Since the marginal at the edge of the data set contains so much noise it

becomes much less precise and, with low utility.

To cope with this problem the research team came up with a method to preprocess data called "K-aggregation". The steps of this method are as the following: Step 1: Two parameters will have to be set in advance. First a threshold will need to be chosen to examine the maximum acceptable error percentage between original dataset and synthetic dataset. Parameter k can be calculated from the maximum acceptable error percentage as stated in previous paragraph. Step 2: After the parameter has been chosen, the original dataset will be put through the DPTable process, from which the synthetic dataset will be generated. Step 3: Synthetic dataset will be compared to the original. If the maximum error across all possible attribute values between the counts in original and synthetic dataset is larger than the error threshold defined in step 1, we will proceed to step 3. Otherwise the synthetic dataset is accepted as usable. Step 4: Since the error is larger than the threshold, we assume that the data value count at the "edge" of the dataset needs to be aggregated to increase due to normal distribution. We will scan the database from the largest and smallest data value and aggregating the counts until the accumulate count exceeds k. In the original table those data value will be replaced with a new value calculated from the weighted mean from the data value. Step 5: If there are multiple attributes presents, step 1~4 can be iterated through all the attributes.

Please take Figure 7 as an example. The algorithm start to scan data from the two sides of the data set, and if the count of a certain value is below the threshold set, it will be combined with the count of the next value. After the threshold was reached the older values will be combined as a weighted new value. After K-aggregation the extremely low counts were combined and more precise counts are possible. To get an idea about the effect of K-aggregation, we also use the artificial data set as an example. In Figure 8, the top chart gives us an idea about the higher error % that the edge cases produce, and it is clear that the cases at the center of the chart produce much lower error %. The chart at the bottom represent shows the data entry count for each weight value.

We process the attribute "weight" in this data set with K-aggregation and have the threshold set to k = 200. In Figure 9. We can see that when the data attribute is pre-processed with K-aggregation, the error % of the counts toward the edge of the data set remains at a much lower level. And in the chart at the bottom we can see that at the edge of the chart the counts are aggregated and given a new value from weighted average of the original values.

To sum it up, K-aggregation can be used to reduce the error % at the edge of a DPTable processed data set, and this also applies to tabular-formatted IoT data sets.

Differential privacy as an option to transfer IoT data securely

Besides releasing sensitive data with privacy protection, differential privacy can also be used to transfer data securely. IoT devices collect information from all kinds of information and send them through Wifi information to remote servers, so it is always possible that someone intercepts those information. If the purpose of data transmission is for further aggregated analysis, differential privacy can come in handy.

Google RAPPOR [14] use differential privacy as a provable mechanism to protect the privacy of transmitted data. When a value is to be transmitted by RAPPOR, its true value will first be converted to binary format, and then passed through a bloom filter. After that the value will then be randomized but "memorized", so that when the value is sent again in the future, this particular randomized value will always represent the same value. And lastly before the values were sent to a remote server the value is randomized again. The remote server will aggregate all those data received and perform statistical estimation regarding how many times a particular string is received. Following this process, one can send carefully randomized information to a remote server for statistical analysis without worrying someone intercepts the data sent. As there are no encryption or decryption involved, there is no risk of leaking a key to an advisory. There're also following up works on RAPPOR to eliminate the need of having to build a dictionary first before data transmission and decoding [15].

4. Conclusion

During the past 10 years research of data anonymity/de-identification has been progress steadily. K-anonymity and differential privacy have been examined extensively to gauge their usefulness in a real world scenario, and the latter has started to be used in some main stream consumer products. In this research we introduced how de-identification techniques can be used for privacy preserve data release and data transmission in an IoT setting. Those techniques can also be used for non IoT purposes, but de-identification techniques, due to its lower requirement for processing power than some of the more sophisticated encryption/decryption schemes, are especially suitable for IoT applications.

Acknowledgement

This research is supported by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 103-2221-E-001-028-MY3 and MOST 106-2923-E-001-001.

References

[1] J. Zhu, K. Zeng, K. Kim, P. Mohapatra, "Improving crowd-sourced Wi-Fi localization systems using Bluetooth beacons." In Proceedings of Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on, 290-298. 2012.

[2] S. S. Chawathe, "Low-latency indoor localization using bluetooth beacons." In Preceedings of 2009 12th International IEEE Conference on Intelligent Transportation Systems, 1-7. 2009.

[3] S. Jamil, A. Basalamah, A. Lbath, "Crowd-sensing traces using bluetooth low energy (BLE) proximity tags." In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, 71-74. 2014.

[4] H. Koyuncu, S. H. Yang, "A Survey of Indoor Positioning and Object Locating Systems." International Journal of Computer Science and Network Security (IJCSNS) 10, 5: 121-128. 2010.

[5] S. S. Chawathe, "Beacon Placement for Indoor Localization using Bluetooth." In Preceedings of 2008 11th International IEEE Conference on Intelligent Transportation Systems, 980-985. 2008.

[6] L. Sweeney, "k-anonymity: a model for protecting privacy," International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5): 557-570, 2002.

[7] Li, N., Li, T., "t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity," Proceedings of the 23nd International Conference on Data Engineering, 2007.

[8] Machanavajjhala, A., Kifer, D., Gehrke J., Venkitasubramania, M., "I-Diversity: Privacy Beyond k-Anonymity," Proceedings of the 22nd International Conference on Data Engineering.(ICDE) , pp.24-35, 2006.

[9] Rubin D. B. Discussion: Statistical Disclosure Limitation, Journal of Official Statistics, Vol. 9, No. 2, pp 461-468, 1993.

[10] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava and X. Xiao, "PrivBayes: private data release via bayesian networks," SIGMOD '14 Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 1423-1434, 2014.

[11] R. Chen, Q. Xiao, Y. Zhang and J. Xu, "Differentially Private High-Dimensional Data Publication via Sampling-Based Inference," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,pp. 129-138, 2015.

[12] B. C. Tai, S. C. Li, Y. Huang, "K-aggregation: Improving Accuracy for Differential Privacy Synthetic Dataset by Utilizing K-anonymity Algorithm", to be presented at AINA 2017.

[13] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez and S. Martínez, "Enhancing data utility in differential privacy via micro-aggregation-based k-anonymity," The VLDB Journal, vol. 23, issue 5, pp. 771-794, October 2014.

[14] Ú. Erlingsson, V. Pihur, A. Korolova, "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response." CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.

[15] G. Fanti, V. Pihur, Ú. Erlingsson, "Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries", Proceedings on Privacy Enhancing Technologies, (3):1–21 2016.

[16] Dwork C., "Differential Privacy: A Survey of Results," in Theory and Applications of Models of Computation Volume 4978 of the series Lecture Notes in Computer Science, pp. 1-19, April 2008.G. Smith, "Paper Title" (to be published).

[17] Gartner Website, Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015, http://www.-gartner.com/newsroom/id/3165317, retrieved on Jan. 26th, 2017.

[18] Andy Greenberg, Apple's 'Differential Privacy' is about collecting your data, but you're your data, https://www.wired.com/-2016/06/apples-differential-privacy-collecting-data/, retrieved on Feb. 6th, 2017.