

A Thesis for the Degree of Ph.D. in Engineering

Data transaction infrastructure
for safe and flexible sharing of private information

February 2019

Graduate School of Science and Technology
Keio University

Yuichi NAKAMURA

Abstract

Thanks to popularization and advancement of information and communication technology, huge and various data have been created, transacted, and stored in human communities. Although only the members of each community use the data, the data would also be worthy to be used to the other purposes of third parties. However, the data cannot be shared with third parties without any restrictions due to privacy concerns. The restrictions have to be defined while concerning requirements of the third party, i.e. a data user, because the restrictions degrade the value of data. The definition is required for each transaction of data sharing since both restrictions and requirements are different among the transactions.

A data transaction infrastructure for safe and flexible sharing of private information (DTI4SFS) is proposed in this study to achieve data sharing from data providers to data users while resolving the problems. DTI4SFS uses anonymization techniques to prevent identification of a record of a specific person for safety while focusing on data sharing. For the flexibility, DTI4SFS allows both data providers and data users to claim their requirements regarding the data sharing by creating either publishing or request rules. The detail of DTI4SFS is proposed in Section 3 while describing the background and the related works in Section 1 and 2, respectively.

Three methods are also proposed in this study to enhance the ability of DTI4SFS. In Section 4, a hardware implementation of an anonymizer is proposed to overcome the bottleneck of the entire flow of DTI4SFS. The proposed anonymizer showed 82.3% reduction in circuit size compared to existing anonymizer based on TCAM. The maximum throughput of the anonymizer was 8.75Gbps. The throughput of the proposed anonymizer is enough to anonymize data at a line speed of OC48 or faster until 8.75Gbps. Additionally, in Section 5, an anonymization method using a self-organizing map to share time-series data is proposed so that DTI4SFS can share time-series data. The proposed anonymization method achieves the same level of k-anonymity with small information loss compared with other conventional method that each data provider anonymizes aggregated data without data sharing among data providers. Information loss was reduced up to 22%. Moreover, a watermarking method for anonymized data is proposed in Section 6 to suppress unauthorized republishing from malicious data users. The proposed watermarking method consists of AES, turbo code, and a gray code converter to protect watermarking information from collusion attacks and distortion attacks. The experimental results showed that the proposed watermarking method could extract more than 95% of the watermarked information from anonymized data even if the data is combined with other records whose size is 95% of the data. Moreover, the embedding of watermarked information can be detected from data that is 30% of the anonymized data. Section 7 concludes this study with the future outlook of this study.

Table of contents

1	Introduction	1
1.1	A large amount of useful data	1
1.2	Secondary use of data	2
1.3	Privacy preservation	2
2	Related works	5
2.1	Privacy-Preserving Data Mining (PPDM)	5
2.1.1	PPDM using cryptography	6
2.1.2	PPDM using perturbation	6
2.1.3	PPDM by restricting database queries.....	7
2.1.4	PPDM using anonymization	8
2.2	Privacy-Preserving Data Publishing (PPDP)	8
2.3	Privacy metrics	9
2.3.1	Definition of technical terms.....	9
2.3.2	k -anonymity [24] [25].....	9
2.3.3	l -diversity [26].....	11
2.3.4	t -closeness [27].....	12
2.4	Information loss.....	12
2.5	De-anonymization	13
2.6	Conventional studies of data sharing and these drawbacks	13
2.7	Objective of this study	15
3	Data transaction infrastructure	17
3.1	Concept of DTI4SFS	17
3.1.1	Publishing rule	17
3.1.2	Request rule.....	18
3.2	One-directional anonymization	18
3.3	Design of DTI4SFS	19
3.3.1	Original data storeroom organization	20
3.3.2	Anonymizing rules storeroom organization.....	20
3.3.3	Data anonymizing and publishing organization	20
3.3.4	Published data storeroom organization	21
3.4	XML-based anonymization sheets	22
3.5	Further studies for DTI4SFS	24
4	Hardware implementation of anonymizer	26
4.1	Existing studies and problem definition.....	26

4.2	Proposed architecture	27
4.2.1	Overview of the proposed architecture.....	27
4.2.2	RAM and buffer.....	28
4.2.3	Controller.....	29
4.2.4	Data generator	30
4.2.5	Hash.....	31
4.2.6	Bloom filter	31
4.3	Details of the anonymization process	32
4.3.1	State transition.....	32
4.3.2	Anonymization process.....	33
4.4	Evaluation of the anonymizer.....	35
4.4.1	Information loss ratio	35
4.4.2	Circuit size	38
4.4.3	Throughput.....	39
4.5	FPGA implementation	40
4.6	Summary.....	41
5	Anonymization method to share electricity usage data	42
5.1	Conventional approaches of PPDM for smart cities.....	45
5.1.1	Approach based on homomorphic encryption.....	45
5.1.2	Perturbation based approach.....	45
5.1.3	Anonymization based approach.....	45
5.2	Requirements for the anonymization	46
5.3	Self-organizing map (SOM).....	46
5.4	Time series analyses	47
5.5	Proposed method to share electricity usage data.....	49
5.5.1	Procedure to acquire shareable data	49
5.5.2	Forecast of peak electricity usage	52
5.6	Evaluation.....	52
5.6.1	Information loss	52
5.6.2	Accuracy of value while considering DR	54
5.7	Summary.....	55
6	Watermarking method for anonymized data	57
6.1	Related techniques of the proposed watermarking method	58
6.1.1	Advanced encryption standard.....	58
6.1.2	Turbo code	63
6.1.3	Gray code converter.....	64
6.2	Proposed watermarking method.....	64
6.2.1	Features of the proposed watermarking method.....	64

6.2.2	Entire flow of the proposed watermarking method	66
6.2.3	Watermarking process of the proposed method.....	67
6.3	Specification of the implemented turbo code.....	70
6.3.1	Success rate of error correction.....	70
6.3.2	Validity of the maximum limit of iteration.....	72
6.3.3	Validity of aborting iterations of internal decoding	73
6.4	Evaluation.....	74
6.4.1	Tolerance against distortion attacks	75
6.4.2	Effectiveness of use of gray code.....	77
6.4.3	Availability of the proposed watermarking method in the proposed infrastructure.....	77
6.4.4	Information loss due to watermarking.....	78
6.5	Summary.....	79
7	Conclusion.....	80

Figures

Figure 1-1 The percentage of American adults and internet-using adults who use at least one social networking site (refer to [1])	1
Figure 1-2 Annual size of data in the world (refer to [3])	2
Figure 2-1 Overview of PPDM using cryptography (Refer to [15])	6
Figure 2-2 Overview of PPDM using perturbation (Refer to [15])	7
Figure 2-3 Overview of PPDM using anonymization (Refer to [15]).....	7
Figure 3-1 Overview of the proposed infrastructure.....	17
Figure 3-2 Design of the proposed infrastructure	20
Figure 3-3 D-XAS example (extract).....	22
Figure 3-4 P-XAR example	23
Figure 3-5 R-XAR Example.....	24
Figure 3-6 Unauthorized republishing in the proposed infrastructure	25
Figure 4-1 Architecture of the proposed anonymizer	28
Figure 4-2 State diagram of the controller.....	32
Figure 4-3 Pseudo code for the anonymization process	34
Figure 4-4 Relationship between information loss ratio and the window size of four patterns of l -diversities (TCAM)	36
Figure 4-5 Relationship between information loss ratio and the window size of four patterns of l -diversities (RAM).....	36
Figure 4-6 Information loss ratio ($k = 2, l = 2$)	37
Figure 4-7 Information loss ratio ($k = 5, l = 5$)	37
Figure 4-8 Information loss ratio ($k = 10, l = 10$)	38
Figure 4-9 Information loss ratio ($k = 20, l = 20$)	38
Figure 4-10 Throughput of the anonymizers	40
Figure 4-11 Xilinx KC705 Kintex-7 evaluation board with FPGA.....	41
Figure 5-1 Electricity usage disclosed from electricity usage data (1) [54]	43
Figure 5-2 Electricity usage disclosed from electricity usage data (2) [55]	44
Figure 5-3 Overview of supposed situation in Section 5.....	44
Figure 5-4 Self-organizing map	47
Figure 5-5 Flow of the proposed method to share electricity usage data.....	50
Figure 5-6 Example of the normal distribution function <i>at</i> ($p = 30, T = 48$).....	50
Figure 5-7 Amount of mean absolute error (<i>MAE</i>) in all and peak time in <i>DK</i> while shifting parameter of variance (σ^2) is from 0.1 to 100.0	55
Figure 5-8 Amount of mean absolute error (<i>MAE</i>) in all and peak time in <i>DI</i> while shifting parameter of variance (σ^2) is from 0.1 to 100.0	55
Figure 6-1 Supposed usage of watermarking in the proposed infrastructure.....	57
Figure 6-2 Encryption flow of CBC mode (refer to [76])	59

Figure 6-3 Decryption flow of CBC mode (refer to [76])	59
Figure 6-4 Encryption flow of CFB mode (refer to [76]).....	60
Figure 6-5 Decryption flow of CFB mode (refer to [76])	60
Figure 6-6 Encryption flow of OFB mode (refer to [76])	61
Figure 6-7 Decryption flow of OFB mode (refer to [76])	61
Figure 6-8 Encryption flow of CTR mode (refer to [76]).....	62
Figure 6-9 Decryption flow of CTR mode (refer to [76])	62
Figure 6-10 Flow of the encoding process of turbo code (refer to [79])	63
Figure 6-11 Puncturing process	64
Figure 6-12 Flow of the decoding process of turbo code (refer to [79])	65
Figure 6-13 Process of recovering from punctured string	65
Figure 6-14 Converting examples of gray code	66
Figure 6-15 Entire flow of the proposed watermarking method.....	67
Figure 6-16 Trellis diagram of systematic convolutional code	68
Figure 6-17 Interleaving process of implemented interleaver	68
Figure 6-18 Success rate of error correction (1).....	70
Figure 6-19 Success rate of error correction (2).....	71
Figure 6-20 Distribution of number of decoding (128 bits).....	71
Figure 6-21 Distribution of number of decoding (256 bits).....	72
Figure 6-22 Distribution of number of decoding (384 bits).....	72
Figure 6-23 Success rate of the decoding process of the three patterns of the maximum limit (1,000, 2,000, and 3,000 times)	74
Figure 6-24 Tolerance against deleting attack	75
Figure 6-25 Tolerance against adding attack	76
Figure 6-26 Tolerance against replacing attack.....	77
Figure 6-27 Comparison of success rates of the error correction between using and not using gray code	78
Figure 6-28 Comparison of Information loss among abstraction methods.....	79

Tables

Table 2-1 Data table of $k = 2$	10
Table 2-2 Anonymized data table of $k = 3$ anonymized from Table 2-1.....	10
Table 2-3 Another example of anonymized data table of $k = 3$	11
Table 2-4 Calculation example of information loss ratio	14
Table 2-5 Anonymized data table of $k = 3$ which sensitive attribute is GID	14
Table 2-6 Data table generated from Table 2-2 and Table 2-5	14
Table 3-1 Example of medical record ($k = 1$).....	18
Table 3-2 Anonymized medical record generated from Table 3-1 ($k = 2$)	18
Table 3-3 Anonymized medical record generated from Table 3-1 (1) ($k = 3$).....	19
Table 3-4 Anonymized medical record generated from Table 3-2 (2) ($k = 3$).....	19
Table 4-1 Data fields of RAM.....	29
Table 4-2 Data fields of buffer.....	29
Table 4-3 Example of masking in the proposed anonymizer	29
Table 4-4 Controller commands	30
Table 4-5 Values of status_an bus and their descriptions	30
Table 4-6 Values of sts_out bus and their descriptions	30
Table 4-7 Circuit utilization	39
Table 4-8 Implementation results	41
Table 5-1 Rate of information loss (<i>RIL</i>) in <i>DK</i> when <i>np</i> and <i>k</i> are set from 2 to 4 and 2 to 10, respectively (MapSize is 5×5)	53
Table 5-2 Rate of information loss (<i>RIL</i>) in <i>DK</i> when <i>np</i> and <i>k</i> are set from 2 to 4 and 2 to 10, respectively (MapSize is 10×10).....	53
Table 5-3 Rate of information loss (<i>RIL</i>) in <i>DI</i> when <i>np</i> and <i>k</i> are set from 5 to 15 and 2 to 20, respectively (MapSize is 10×10).....	53
Table 5-4 Rate of information loss (<i>RIL</i>) in <i>DI</i> when <i>np</i> and <i>k</i> are set from 5 to 15 and 2 to 20, respectively (MapSize is 20×20).....	53
Table 6-1 Example data table after grouping process	69
Table 6-2 Example data table after modification process.....	69
Table 6-3 Rate of executions that were unstable until the maximum limit.....	73
Table 6-4 Rates of incorrect aborting.....	73

Abbreviations

A

ADF	Augmented Dickey–Fuller (test)	P50
AES	Advanced Encryption Standard	P55
AIC	Akaike’s Information Criterion	P50
AR	AutoRegressive (model)	P45
ARIMA	AutoRegressive Integrated Moving Average (model)	P46
ARMA	AutoRegressive Moving Average (model)	P46
ARS	Anonymizing Rules Storeroom organization	P18
ASIC	Application Specific Integrated Circuit	P25

B

BMU	Best Matching Unit	P44
-----	--------------------------	-----

C

CBC	Cipher Block Chaining (mode)	P55
CFB	Cipher–FeedBack (mode)	P55
CRC	Cyclic Redundancy Check	P29
CSS	Cascading Style Sheets	P22
CTR	CounTeR (mode)	P55

D

DAP	Data Anonymizing and Publishing organization	P18
DR	Demand Response	P41
DTI4SFS	Data Transaction Infrastructure for Safe and Flexible Sharing of private information	P14
D–XAS	Data in XAS format	P18

E

EMD	Earth Mover’s Distance	P11
-----	------------------------------	-----

F

FHE	Fully Homomorphic Encryption	P5
FPGA	Field Programmable Gate Array	P24

G

GDPR	General Data Protection Regulation	P2
GID	Group Identifier	P8

I

IL	Information Loss	P11
IoT	Internet of Things	P1
IV	Initial Vector	P56

M

MA	Moving Average (model)	P45
MAE	Mean Absolute Error	P51

N			
	NDA	Non-Disclosure Agreement	P12
O			
	ODS	Original Data Storeroom organization	P18
	OFB	Output-FeedBack (mode)	P55
P			
	PDS	Published Data Storeroom organization	P19
	PKI	Public Key Infrastructure	P19
	PPDM	Privacy-Preserving Data Mining	P4
	PPDP	Privacy-Preserving Data Publishing	P7
	P-XAR	Publishing rule in XAR format	P18
	P-XAS	Published XAS	P19
R			
	RAM	Random Access Memory	P24
	R-XAR	Request rule in XAR format	P18
S			
	SARIMA	Seasonal AutoRegressive Integrated Moving Average	P45
	SNS	Social Networking Service	P1
	SOM	Self-Organizing Map	P41
	SOVA	Soft Output Viterbi Algorithm	P61
	SSL	Secure Sockets Layer	P19
T			
	TCAM	Ternary Content Addressable Memory	P24
W			
	WSOM	Weighted SOM	P47
X			
	XAR	XML-based Anonymization Rules	P18
	XAS	XML-based Anonymization Sheets	P18
	XML	Extensible Markup Language	P20

1 Introduction

1.1 A large amount of useful data

Thanks to advancement and popularization of information and communications technologies, a large amount of data called big data have been generated, dealt, and stored by various organizations such as institutes, governments, and companies. The data may contain customer data of companies and location data of trucks. Also, the size of the stored data is massive. For instance, the number of users of social networking services (SNSs) increases in the past decade. According to [1], 65% of American adults were SNS users in 2015 whereas the percentage was 7% in 2005 (Figure 1-1). Amount of data of the users is so large that Scuba, which is a data management system in Facebook, stores around 70 TB of compressed data for real-time analysis [2]. Additionally, according to [3], annual size of data in the world was estimated to be 33 ZB in 2018. The size is forecasted to increase to 175 ZB by 2025 (Figure 1-2).

Emerging of Internet of Things (IoT) also enables aggregating and storing various types of data. IoT devices are used in various fields such as services in a smart city [4], industrial systems [5], and health care services [6]. The data contain sensor data such as electricity usage, room temperature, and location information. Also, actuators of IoT devices periodically send their status and feedback of control such as door-lock status and the current setting of an air conditioner. These data are periodically sent from the various devices to a data aggregator.

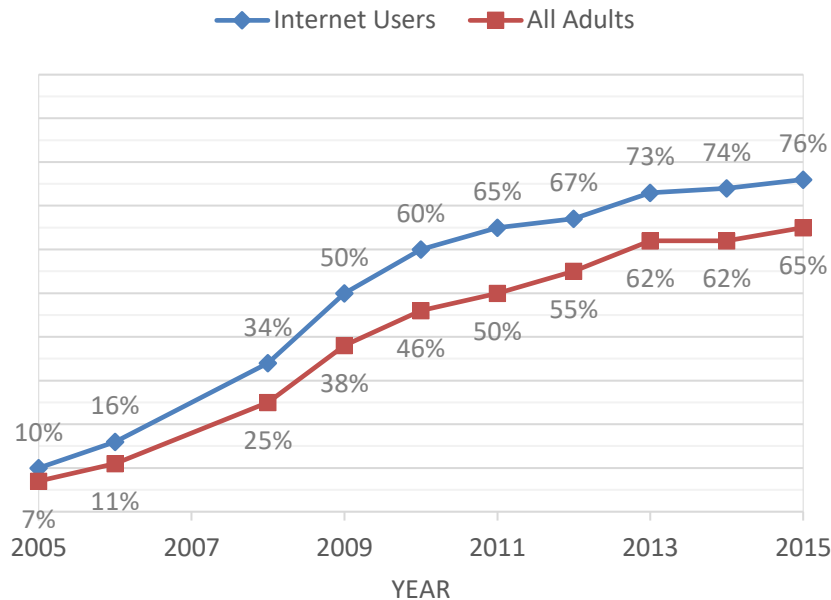


Figure 1-1 The percentage of American adults and internet-using adults who use at least one social networking site (refer to [1])

1 Introduction

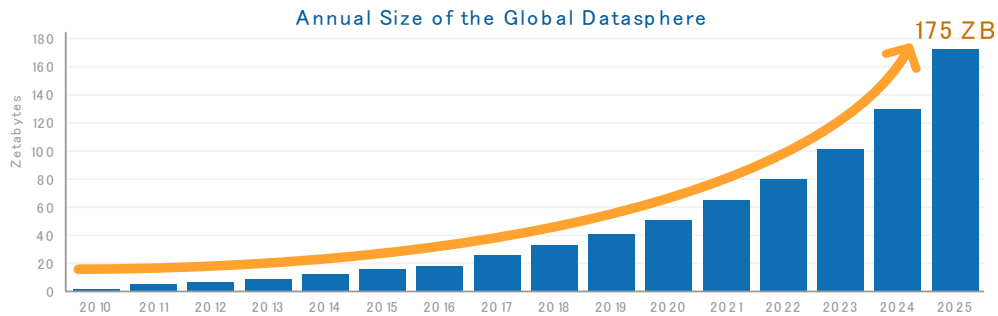


Figure 1-2 Annual size of data in the world (refer to [3])

1.2 Secondary use of data

The stored data are useful for secondary use. Secondary use is a data utilization that purpose is different from the primary purpose when storing data. For instance, electricity usage data measured by a smart meter can be used to shift peak demand of electric power whereas the data have been originally measured to calculate power bills. In this example, the peak shift is the secondary use.

Secondary use among different fields is expected as well as a secondary use in a single field [7]. There is an idea named open data to share data beyond the boundary of fields. According to [8], open data and content can be freely used, modified, and shared by anyone for any purpose. One of associations that promote open data is a government. Several governments of countries including the United States (US) and Japan provide a website to share data with citizens [9] [10]. The movement is called open government. Citizens can read the data and release applications that use the shared data. The main contents of the applications are sightseeing, disaster prevention, traffic, and weather because the government has originally collected the main portion of the shared data to take a census.

1.3 Privacy preservation

A data publisher has to consider private information regarding published data from the viewpoint of privacy protection. Private information means that information which is important for the source of information such as a data provider, and information which the data provider wants to be secret. Private information may contain an address, a result of medical check, and information about his/her daily life. Currently published data for open data do not include private information because the data are statistical data containing a large amount of data records. In other words, an attacker cannot detect information regarding a specific person from such the statistical data. In the future, more specific data, such as electricity usage data captured from houses in smart cities, would be shared as open data thanks to the popularization of IoT technology. However, data publishers have to consider the data privacy because the data may contain private information. In other words, the data must be modified to protect containing

1 Introduction

private information.

For preserving private information safely, several schemes have been implemented by countries and unions. In 1995, the European Union (EU) implemented Data Protection Directive that official name is Directive 95/46/EC [11]. In the directive, a country which fulfills regulations regarding privacy protection only be approved to receive data containing private information from the EU. Especially, the EU made an agreement with the US that was named Safe Harbor Agreement in 2001. The agreement invalidated and replaced with an updated version named EU-US Privacy Shield in 2015 and 2016, respectively [12]. However, the content of the Data Protection Directive does not fit to the Internet nowadays because the directive was implemented at the very beginning of the Internet. To replace the directive, the EU had adopted and implemented the General Data Protection Regulation (GDPR) in 2016 and 2018, respectively [13]. GDPR requires several restrictions to a foreign company which deals with data of the EU to protect the data as same as the previous directive while the requirements were updated. Since international companies have customers in the EU, GDPR would be a de facto global standard. The Japanese government also have implemented an act on the protection of personal information. The original version of the act had implemented in 2005, and an amended version was released in 2017 [14]. The amended version defines personal information as data regarding people which allows identifying a record of a specific person even if the identification can be achieved using other data that can be referred easily. Both GDPR and the amended act in Japan require to protect personal information from data leak, and breached companies would be punished severely.

The schemes described in the previous paragraph must be considered to share data containing private information. One of the ways to the sharing is using anonymization. Anonymization is a name of techniques that generalize information in data by modifying values of the data. The modification prevents identifying a specific person's record. Namely, anonymization protects private information from data sharing.

GDPR defines that anonymized data are not required to be controlled under GDPR. The amended version of the Japanese act on the protection of personal information also defines that anonymized data are not personal information if both a data anonymizer and a data user fulfill conditions to protect regarding private information.

Sharing of data containing private information can be achieved using anonymization if the sharing fulfills conditions required by the schemes described above. There is a technical condition for a data anonymizer required by the Japanese act. The condition is an appropriate modification of anonymization. According to the definition of the act, a data value that makes a record unique must be deleted or generalized. The data value includes name, id, and outlier.

Moreover, there is another condition that prohibits trying de-anonymization. This condition is also required for a data user. Anonymization is capable of fulfilling the conditions by selecting appropriate anonymization scheme and items of anonymized data.

For achieving data sharing considering data privacy, in this study, a data transaction infrastructure for safe and flexible sharing of private information is proposed. The proposed

1 Introduction

infrastructure focuses on privacy preservation of data but also data transaction for several reasons. One of the reasons is to prevent de-anonymization by combining previously published anonymized data. The other reason is to consider requirements regarding data sharing claimed by both a data provider and a data user. This consideration provides flexibility of data sharing. The proposed infrastructure focuses on safe data sharing as long as flexibility. Safe data sharing means that private information of shared data is preserved by using anonymization including de-anonymization. Additionally, the proposed infrastructure considers how to suppress the re-publishing of shared anonymized data by proposing watermarking of anonymized data.

This paper is organized as follows: Related works of the proposed infrastructure and drawbacks of conventional studies are described in Section 2. The overview and contribution of this study are also described in the end of this section. The detail of the proposed infrastructure is described in Section 3. This study contains three further proposals to enhance the ability of the proposed infrastructure. The proposals, which are an anonymizer for high-speed anonymization, anonymization method for time-series data, and watermarking method for anonymized data, are proposed and evaluated in Section 4, 5, and 6, respectively. Section 7 concludes this study.

2 Related works

2.1 Privacy–Preserving Data Mining (PPDM)

Private information in data has to be preserved when the data are used. Namely, disclosing of private information must be prevented for the preservation. On the other hand, such data are valuable for data mining to improve relevant services. Even though information acquired by data mining does not contain private information, data mining has a risk of disclosing private information because data mining often uses data containing private information. Techniques to mine data while preserving private information are called Privacy–Preserving Data Mining (PPDM) [15] [16] [17] [18].

Here is an example of PPDM for marketing. Information on membership card of supermarkets includes shopping history of customers. The information could be valuable for each supermarket for marketing, and also valuable for a maker of goods sold at supermarkets to support the marketing or developing new products. There is two private information requiring preservation. One of the information is the shopping history of each customer. Some techniques such as anonymization can preserve this information. The other one is a data source, i.e., which supermarket provided the data. This data source information should be preserved because if a maker knew the data source, the maker could maliciously put a specific supermarket into a disadvantageous situation. One of the ways to keep data source secret against the maker is to make relationship among supermarkets. If the data source was hidden or generalized appropriately before providing the data to the maker, the data source would be preserved. Purpose of PPDM is to obtain knowledge while nobody gets disadvantages. In this example, the maker obtains valuable knowledge while preserving the privacies of both shoppers and supermarkets. Other examples are briefly shown as follows.

- Trend analysis of symptoms using medical records while preserving information of related patients and hospitals.
- Analysis of client lists of loan to detect a client who would be a bankrupt individual. Privacy of clients in the lists and consumer finance companies that provided the lists are preserved in the analysis process.
- Referring to a client list of a company to check whether a suspect is listed in the list. The referring is processed while both a suspect list and the client list are not published.

In March 2004, a workshop regarding PPDM entitled DIMACS/PORTIA Working Group Meeting on Privacy–Preserving Data Mining [19] was held in the US. According to the workshop, PPDM studies mainly used public–key encryption. In contrast, various approaches have been proposed nowadays. Typical methods use either encryption, perturbation, restriction of database

2 Related works

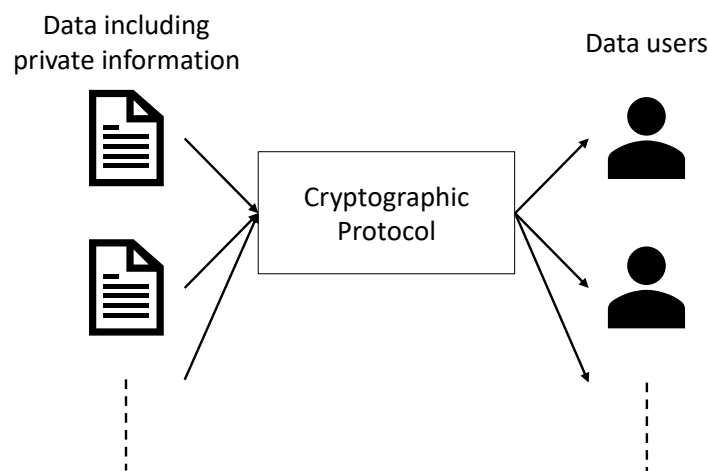


Figure 2-1 Overview of PPDM using cryptography (Refer to [15])

queries, and anonymization.

2.1.1 PPDM using cryptography

In PPDM using cryptography, data mining is processed while keeping private information encrypted (Figure 2-1). The calculation of data mining is divided into simple sub-calculations that can be processed using a special calculation tool. The tool allows to calculate data while keeping the data, calculation, and calculation result encrypted. Since each sub-calculation is encrypted, the whole calculation is also encrypted. The drawback of the PPDM is calculation cost. Since every sub-calculation requires a process of cryptography and network communication, the PPDM is generally not suitable for a large amount of data.

Fully Homomorphic Encryption (FHE) is one of the techniques of PPDM using encryption [20] [21] [22]. FHE allows calculating data while keeping related information encrypted. The information contains not only the data but also equations of the calculation. Thanks to this feature, data miner can ask an untrusted calculation agency to analyze data while keeping data, calculation result, and purpose of the data mining. Although FHE is useful to protect private information against divulging of information, safe data sharing cannot be achieved by using FHE only because FHE is not the technique to obscure the content of the information for data privacy.

2.1.2 PPDM using perturbation

Perturbation can be used to achieve PPDM (Figure 2-2). This PPDM has two steps. In the first step, noise is added into values in data before data mining. The noise of each value is generated by a perturbation function. A data miner analyzes the modified data while referring to the probability distribution of the function to estimate actual values of the data in the next step. Private information in data is preserved through the two steps because the original data is not provided to the data miner.

Additionally, this PPDM has an advantage in terms of calculation cost compared with other

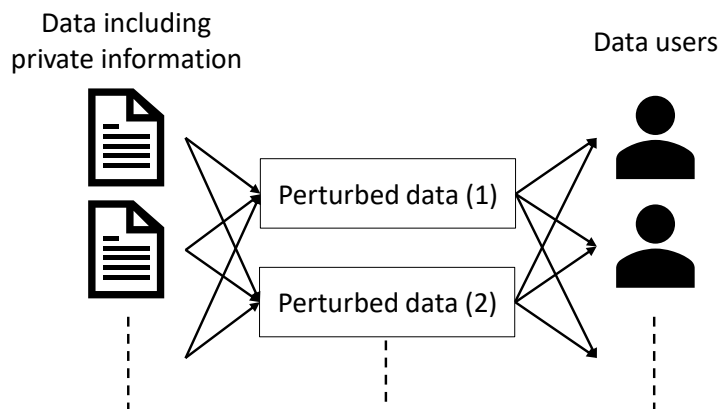


Figure 2-2 Overview of PPDM using perturbation (Refer to [15])

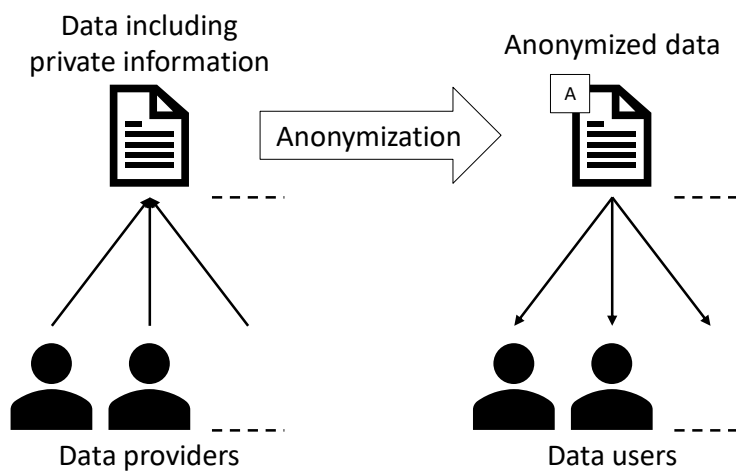


Figure 2-3 Overview of PPDM using anonymization (Refer to [15])

PPDMs. This PPDM, therefore, suits for a large amount of data. However, using perturbation has a drawback regarding the guarantee of privacy preservation. Private information is preserved from the viewpoint of statistics only. Namely, the robustness of the preservation depends on dedicated perturbation function.

2.1.3 PPDM by restricting database queries

Privacy information in a database can be preserved when queries to the database are restricted appropriately. For example, an outlier should not be published because the related person of the outlier may be identified. This information will be preserved if only data property such as maximum value, minimum value, and average are accessible.

The lack of this PPDM is a restriction of data access. Data miners cannot access data as he/she prefers. The rules of the restrictions have to be defined while considering the usability of the accessible information.

2 Related works

2.1.4 PPDM using anonymization

PPDM can be achieved when anonymization is used (Figure 2–3) [23]. Anonymization preserves private information by generalizing values in data. There are some techniques of the generalization such as replacing, masking, and micro aggregation. The replacing is a technique to replace an original value with its abstracted value. For example, “apples” could be replaced with “fruits.” Some types of values such as age can be generalized by the masking a more sensitive part of the value. “25 years old” can be generalized to “2* years old” by using a masking character “*.” The micro aggregation firstly groups records along with similarity of values in each record. Then, values in each group are replaced with the same representative value such as an average. Perturbation described in Section 2.1.2 can be regarded as one of the generalization techniques. Anonymization guarantees irreversibility when there are no references to obtain an original value of anonymized data.

Some privacy metrics to indicate anonymity level have been proposed such as k -anonymity [24] [25], l -diversity [26], and t -closeness [27]. Each metric represents the uniqueness level of a record in data from each other’s different viewpoint.

Anonymization is adaptable for Privacy-Preserving Data Publishing (PPDP) as well as PPDM in contrast to other PPDM techniques because of two advantages as follows. One of the advantages is that anonymization generalizes data values. Since a modified value holds correct information whereas the value is generalized, a data user can obtain correct knowledge. Here is the other advantage that the anonymity level can be represented by the metrics. Thanks to the metrics, data providers can define the strength of privacy preservation clearly. Additionally, data users can know how much the anonymized data was generalized before obtaining the data.

2.2 Privacy-Preserving Data Publishing (PPDP)

Techniques to publish data while preserving data privacy are named PPDP [23] [28]. Purpose of PPDP is to obtain knowledge while preserving data privacy as same as PPDM. However, PPDP is different to PPDM in terms of purpose. PPDP focuses on how to publish data while preserving data privacy whereas PPDM focuses on how to analyze data while preserving data privacy.

Anonymization and perturbation are typical techniques of PPDP, and anonymization is selected in most cases. As mentioned in Section 2.1.2, the robustness of perturbation techniques is guaranteed only from the statistical perspective. According to [29], data privacy is not preserved enough only by using perturbation in some cases. In contrast to perturbation, a record regarding a specific person is not able to be specified from anonymized data as long as its original data is not referred. Namely, anonymization can preserve data privacy robustly rather than perturbation.

There are many studies of anonymization techniques itself. One of the typical topics is how to suppress the degradation of value due to anonymization while satisfying the anonymity level [30] [31] [32]. Additionally, [33] proposes a method to extract valuable information from anonymized data. [34] focuses on the acceleration of the anonymization process. [35] and [36] try to protect

2 Related works

from attacks on anonymization.

2.3 Privacy metrics

As mentioned in Section 2.1.4, privacy metrics have been proposed. Some anonymization techniques allow controlling anonymity level by using privacy metrics. Additionally, the strength of privacy protection of anonymized data can be assessed based on privacy metrics. In this paper, technical terms regarding anonymization are defined as follows [25] [26].

2.3.1 Definition of technical terms

2.3.1.1 Data table

Data table is a table that consists of information such as a list of customers and a medical record. In this paper, column and row of a data table are termed “attribute” and “tuple,” respectively. Each tuple in the data table holds a record. Each record indicates values regarding a single or group of people such as customers and patients. The record consists of several values that are categorized by attributes. Each attribute describes an item of its column such as birthday.

2.3.1.2 Attribute

Attributes can be categorized according to the level of privacy information. The most critical type of attributes is named “identifier.” The identifier has a value that provides information to identify who is related to the tuple. Such information includes ID number, name, and cell-phone number. Some of the other attributes can be regarded as an identifier when the attribute combines with other attributes. Such attributes are termed “quasi-identifier.” Attributes such as birthday, gender, and group identifier (GID) are categorized to quasi-identifier. A group of tuples that have the same value in every quasi-identifier is termed “q*-block.”

2.3.1.3 Sensitive attribute

An attribute that is important for a data analyst is called “sensitive attribute.” Values in the sensitive attributes are not modified. In contrast to sensitive attribute, an attribute that is acceptable to be modified is named “non-sensitive attribute.”

2.3.2 k -anonymity [24] [25]

According to [24], k -anonymity is defined as follows.

- The data table is said to satisfy k -anonymity if and only if each sequence of values in quasi-identifier in the data table appears with at least k occurrences in the quasi-identifier.

Table 2-1 is an example of the data table. Attributes of Birth, Gender, and GID in the data table are quasi-identifiers while the Problem attribute is a sensitive attribute. For the readability, there are tuple number t . Since t is out of the data table, anonymization for the data table does

2 Related works

Table 2-1 Data table of $k = 2$

	<i>Quasi-identifiers</i>			<i>Sensitive Attribute</i>
	Birth	Gender	GID	Problem
$t1$	1980	male	121	fever
$t2$	1980	male	121	stomachache
$t3$	1980	male	121	headache
$t4$	1980	female	121	headache
$t5$	1980	female	121	stomachache
$t6$	1981	male	125	headache
$t7$	1981	male	125	fever

Table 2-2 Anonymized data table of $k = 3$ anonymized from Table 2-1

	<i>Quasi-identifiers</i>			<i>Sensitive Attribute</i>
	Birth	Gender	GID	Problem
$t1$	1980	male	121	fever
$t2$	1980	male	121	stomachache
$t3$	1980	male	121	headache
$t4$	198*	no data	12*	headache
$t5$	198*	no data	12*	stomachache
$t6$	198*	no data	12*	headache
$t7$	198*	no data	12*	fever

not consider t . There are three q^* -blocks that consist of $t1 \sim t3$, $t4$ and $t5$, and $t6$ and $t7$, respectively. Since the minimum number of tuples in the q^* -block is two, the data table satisfies 2-anonymity. Let me suppose an attacker attempts to know Problem value of a lady, who was born in 1980. In this situation, even though the attacker has the data table, the attacker cannot identify whether the Problem value is a headache or a stomachache from $t4$ and $t5$.

When 3-anonymity is required, $t4 \sim t7$ in Table 2-1 will be further anonymized to Table 2-2. Two q^* -blocks exist in the data table. Each q^* -block consists of at least three tuples that are $t1 \sim t3$ and $t4 \sim t7$.

k -anonymity seems to eliminate the risk to the identification of a record of a specific person. However, there are several attacks to identify information regarding a specific person.

2.3.2.1 Homogeneity attack

An attacker can sometimes identify a value of a record regarding a specific person even if the attacker cannot identify which record the person relates. This attack is named homogeneity attack. Table 2-3 is another example of the anonymized data table. In this data table, $t1 \sim t3$ have the same value (fever) in a sensitive attribute which is Problem attribute. Therefore, an attacker who knows values in quasi-identifiers of Alice is able to identify her value in Problem attribute whereas the attacker cannot identify her tuple.

2.3.2.2 Background knowledge attack

$t4 \sim t7$ in Table 2-3 have another risk of attacks named background knowledge attack. For

2 Related works

Table 2-3 Another example of anonymized data table of $k = 3$

	<i>Quasi-identifiers</i>			<i>Sensitive Attribute</i>	
	Birth	Gender	GID	Problem	
$t1$	1980	female	121	fever	Alice
$t2$	1980	female	121	fever	
$t3$	1980	female	121	fever	
$t4$	198*	no data	12*	poor circulation	Bob
$t5$	198*	no data	12*	poor circulation	
$t6$	198*	no data	12*	headache	
$t7$	198*	no data	12*	headache	

instance, an attacker attempts to obtain Problem value of Bob, and the attacker knows his values of quasi-identifiers. In contrast to the example of Alice, the attacker cannot identify his Problem value because there are two candidates of the value that are poor circulation and headache. However, the attacker can estimate the value if the attacker has background knowledge that men rarely gets poor circulation. Since there are only two candidates and one of the candidates is poor circulation, the attacker can estimate Bob has a headache.

2.3.3 l -diversity [26]

Attacks described in 2.3.2.1 and 2.3.2.2 are effective due to lack of variety of values in each q^* -block. l -diversity has been proposed to overcome the attacks. Here is the definition of l -diversity.

- l -diversity is fulfilled when each quasi-identifier has l or large kinds of values in each q^* -block.

l -diversity of Table 2-1, Table 2-2, and Table 2-3 are two, three, and one, respectively. Since the maximum value of l is restricted by the number of tuples in the q^* -block, the relationship between k and l is represented by the inequation 2-1.

$$k \geq l$$

2-1

Here are more concrete definitions of l -diversity as follows.

2.3.3.1 Entropy l -diversity

Entropy l -diversity has been proposed in [37] while considering homogeneity attacks. When $p(q^*, s)$ denotes the fraction of tuples which have value $s \in S$ as a sensitive attribute in q^* -block, the inequation 2-2 represents the condition to satisfy entropy l -diversity.

$$-\sum_{s \in S} p(q^*, s) \log(p(q^*, s)) \geq \log(l)$$

2-2

2 Related works

2.3.3.2 Recursive (c, l) -diversity [26]

Recursive (c, l) -diversity has been proposed while considering both homogeneity attacks and background knowledge attacks. Let r_i denote the number of times the i th most-frequent sensitive value in a q^* -block in a data table. The q^* -block satisfies (c, l) -diversity when inequation 2-3 is fulfilled in the q^* -block. n in the inequation represents the number of values of the sensitive attribute in the q^* -block. c denotes a constant. The data table satisfies (c, l) -diversity when every q^* -block in the data table satisfies (c, l) -diversity.

$$r_1 < c(\sum_{k=l}^n r_k)$$

2-3

2.3.3.3 l -diversity of multiple sensitive attributes

When a data table has multiple sensitive attributes, the maximum l -diversity in the sensitive attributes is regarded as l -diversity of the data table. This is because when values of sensitive attributes in a tuple regard as a value group, a number of kinds of the groups equals to the maximum l -diversity.

2.3.4 t -closeness [27]

As shown in 2.3.3, l -diversity lets a data table have a variety of values in sensitive values in order to protect data from background knowledge attacks. However, the balance of information entropy among q^* -blocks should be considered for the further protection. In other words, a background knowledge attacker could identify the target tuple easily when a q^* -block which has high information entropy compared with other q^* -blocks. Difference of information entropy among q^* -blocks should be small in order to degrade the threat of background knowledge attacks.

t -closeness has been proposed to enhance protection from background knowledge attacks while considering the balance of information entropy among q^* -blocks. Concretely, t -closeness sets a threshold of a fraction of tuples based on the value of a sensitive attribute in each tuple. For each tuple in every q^* -block, the difference of a fraction of the tuple between the q^* -block and the data table must be smaller than t . Earth Mover's Distance (EMD) [38] is used to calculate the difference.

t -closeness focuses on whole tuples of a data table whereas k -anonymity and l -diversity focus on each q^* -block only. However, t -closeness is not popular compared with the other metrics because controlling fractions of all tuples in a data table imposes high calculation cost.

2.4 Information loss

Anonymization affects information value of data tables because it modifies values in data tables to preserve data privacy. Information Loss (IL) is a metric to represent how much information was lost due to anonymization. When a tuple t in a data table T is defined as $t \in T\{t_1, t_2 \dots t_N\}$

2 Related works

where quasi-identifier in T is represented as $QI_T\{A_1, A_2 \dots A_{N_A}\}$, information loss $IL(T)$ is defined as an equation 2-4. Each quasi-identifier can be generalized up to DGH times, and currently generalized h times. Therefore, $\frac{h}{|DGH|}$ denotes information loss of a value of a quasi-identifier in a tuple. $IL(T)$ denotes average of $\frac{h}{|DGH|}$ in the data table. Both $\frac{h}{|DGH|}$ and $IL(T)$ take a value from zero to one. IL of a data table before anonymization is zero whereas a completely anonymized data table takes $IL = 1$.

$$IL(T) = \frac{\sum_{t_j \in T} \sum_{A_i \in QI_T} \frac{h_{ij}}{|DGH_{A_i}|}}{|T| \cdot |QI_T|} = \frac{\sum_{j=1}^N \sum_{i=1}^{N_A} \frac{h_{ij}}{|DGH_{A_i}|}}{N \cdot N_A}$$

2-4

There are some equations to calculate IL whereas the equation 2-4 is the most typical form. The range of IL in this equation is $0 \leq IL \leq 1$. Such metric is called information loss ratio. Table 2-4 shows a calculation example of information loss ratio when values in the “address” attribute is masked in stages.

Information loss is not able to be used to compare different types of data because this metric represents the loss of information value by measuring the frequency of generalization. In this paper, information loss is used to compare anonymized data that are same originally.

2.5 De-anonymization

Anonymized data have a risk of de-anonymization in PPDM and PPDP. De-anonymization is one of background knowledge attacks to re-identify an original value of the anonymized tuple while referring background knowledge such as private information of a targeted person obtained from his/her posts in SNS. Especially in PPDP, anonymized data that have already been published can be regarded as background knowledge.

Table 2-2 and Table 2-5 are anonymized data tables of 3-anonymity. Both of them were made from Table 2-1 by different anonymization processes. Although each data table satisfies 3-anonymity, a less anonymized data table shown in Table 2-6 appears when the two anonymized data tables are combined. The data table is 1-anonymity when GID is selected to a quasi-identifier.

2.6 Conventional studies of data sharing and these drawbacks

As described in Section 1.2, data sharing facilitates solving of problems and advancement of studies. Data sharing has been especially well studied in the medical field to share study results of state of the art among clinical fields as soon as possible [39] [40] [41].

One of the ways to share data including private information is making a contract between a data provider and a data user to protect private information by legal aspect. The contract is

2 Related works

Table 2-4 Calculation example of information loss ratio

<i>IL</i>	Address		
0	Kohoku-ku	Yokohama-shi	Kanagawa-ken
1/3	*-ku	Yokohama-shi	Kanagawa-ken
2/3		*-shi	Kanagawa-ken
1			*

Table 2-5 Anonymized data table of $k = 3$ which sensitive attribute is GID

	<i>Quasi-identifiers</i>			<i>Sensitive Attribute</i>
	Birth	Gender	Problem	GID
t_1	198*	no data	*	121
t_2	198*	no data	*	121
t_3	198*	no data	headache	121
t_4	198*	no data	headache	121
t_5	198*	no data	*	121
t_6	198*	no data	headache	125
t_7	198*	no data	*	125

Table 2-6 Data table generated from Table 2-2 and Table 2-5

	Birth	Gender	GID	Problem
t_1	1980	male	121	fever
t_2	1980	male	121	stomachache
t_3	1980	male	121	headache
t_4	198*	no data	121	headache
t_5	198*	no data	12*	stomachache
t_6	198*	no data	125	headache
t_7	198*	no data	12*	fever

called the Non-Disclosure Agreement (NDA). Although using NDA enables sharing data without any modifications, this advantage would be a risk of disclosure from malicious third parties. Even if the third party would receive a sanction, it is hard to retrieve the private information in the disclosed data. Therefore, data privacy must be protected by techniques of either PPDM or PPDP as well as legal aspect for the data sharing.

Although techniques such as PPDM and PPDP have been investigated in numerous studies, a method of secure and flexible publishing the data to enable secondary use has not been definitively established. Balamurugan et al. propose a data partitioning method for multiparty computation while considering privacy preservation [42]. A dataset is divided into several parts for multiparty computation. The method maintains the range and size of the parts to avoid the situation that multiple parts of the same data are provided to a single user in order not to provide private information to the user. However, the method cannot use for the data sharing that this study considers because the method does not consider another privacy risk that multiple malicious users combine their provided parts to obtain the private information. Ruvan et al. propose a framework to share data for data mining while preserving data privacy [43]. In the framework, the shared database is anonymized to fulfill k -anonymity given by the database owner. However, the

2 Related works

framework does not meet concerning the requirements of k -anonymity level from data users even though anonymization degrades the data quality. Additionally, the framework is not flexible in terms of selection of privacy preservation techniques because the framework allows using k -anonymity only. [44] discusses how to do PPDM while supposing a data-sharing scenario for data mining with some types of user roles, and introduces ideas of how to make a compromise of privacy-preservation level among the users. However, the proposal does not consider about the framework of communication among users of the supposed infrastructure. Moreover, the paper is lack of attention regarding privacy risk by a transaction of data sharing. The proposed infrastructure in this study overcomes these problems.

As described in Section 1.3, the act on the protection of personal information has been defined in Japan. However, there is a problem when Japanese companies transact data based on the act. The problem is that guidelines for the companies are different among Japanese ministries. Therefore, actions of the companies are different according to their related ministries. Also, the difference could confuse companies which are related to multiple ministries. These guidances must be combined into a single guidance to promote secondary use of data among different fields. The proposed infrastructure could take the role on the unified guidance.

Some people may have a conservative attitude to provide data due to their anxieties about privacy risk even if the privacy-preserving way of the data publishing was well considered. The situation would happen when the entire flow of the data sharing is not sharply defined. Additionally, the anxieties would reduce if people whose privacy are related to data are able to actively manage requirements regarding the data sharing. The proposed infrastructure would solve the problems.

Furthermore, as described in Section 2.5, after calculating and publishing anonymized data from a data source, another anonymized dataset, calculated and published from the same source may cause a privacy information leak if an unauthorized person can access both sets of anonymized data. When calculating and publishing anonymized data, it is necessary to consider all of the previously published data from the same source.

By considering these issues, it is crucial to establish a clear suggestion of technological guidance, an infrastructure, and a technical standard of protocols for the secondary use of data. The development of the protocol and infrastructure is especially important to its development. It will facilitate collaboration between data providers and data users, and thus increase their data publishing activity. It will develop the market for secondary uses of data in conjunction with advanced services such as market research, estimation of a route of infection, and traffic pattern analysis. Moreover, it will reduce the utility costs for both providers and consumers of secondary use data, owing to the unification of data processing procedures.

2.7 Objective of this study

In this study, a data transaction infrastructure for safe and flexible sharing of private information (DTI4SFS) is proposed. DTI4SFS uses anonymization to preserve data privacy. Additionally, a protocol and XML-based data format for the proposed infrastructure are proposed.

2 Related works

DTI4SFS prevents further leaks of private information by employing the previously anonymized data as a publishing history.

This study consists of three further proposals to enhance the transaction ability of DTI4SFS. The two of the three proposals enhance scalability of DTI4SFS from two aspects, respectively. One of the aspects is acceleration of the anonymization process, which is a bottleneck of the entire process flow. If the anonymization process becomes faster, DTI4SFS can transact large amount of data. Additionally, the acceleration enables situations that requires high-speed transaction to provide fresh information to data users. The other aspect of the scalability is variety of data types that DTI4SFS is able to anonymize. This study proposes an anonymization method for time-series data that is popular data type such as sensor data. The last proposal in this study is a watermarking method for anonymized data. This proposal is effective to suppress unauthorized republishing from malicious data users. Although this study focuses on the three proposals, any other technique for safe data sharing is acceptable for DTI4SFS thanks to the flexibility.

The contribution of this study is the proposal of DTI4SFS. The proposed infrastructure provides common protocols and formats for safe data transaction that conventional studies described in Section 2.6 have not focused enough. DTI4SFS would facilitate safe data sharing among different fields for secondary use of data. Additionally, the three further proposals enhance the ability of the proposed infrastructure.

3 Data transaction infrastructure

3.1 Concept of DTI4SFS

DTI4SFS allows secure and flexible data sharing. The security means privacy preservation of private information in shared data. The flexibility means both data providers and data users can require or request their requirements such as contents of the data and the method of privacy preservation including its parameters. DTI4SFS achieves the security based on anonymization techniques. The security focuses not only data modification using anonymization but also data transactions such as multiple publishing of data anonymized from the same data.

DTI4SFS supposes several roles as shown in Figure 3-1. Data providers provide data to DTI4SFS to share the data. When a data provider provides a dataset to DTI4SFS, the data provider creates a publishing rule that defines restrictions regarding the sharing of the data. The publishing rule is also submitted to DTI4SFS. Data users use data provided from DTI4SFS for data analyses for secondary use of the data. A data user creates a request rule when the data user plans to acquire a dataset. The request rule includes requirements of the data user regarding data sharing. The data user requests providing of the required data to DTI4SFS by submitting the request rule to DTI4SFS.

3.1.1 Publishing rule

Data providers can define restrictions regarding their providing as follows.

- Allowable combinations of sensitive attributes: Allowable anonymity level for each combination is also able to be defined.
- Items that can be allowable to be selected as quasi-identifiers: The data provider can define allowable anonymity levels for each quasi-identifier.

Data providers may want to provide some items without any restrictions for reasons such as an advertisement. DTI4SFS allows data providers to select items as such attribute (open attribute).

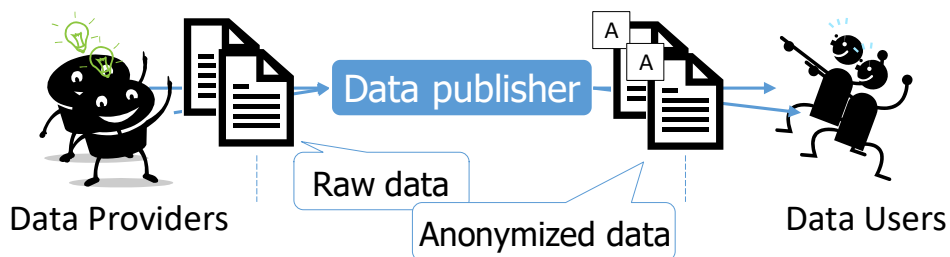


Figure 3-1 Overview of the proposed infrastructure

3 Data transaction infrastructure

Table 3-1 Example of medical record ($k = 1$)

	Birth	Gender	Problem
$t1$	1970	male	cold
$t2$	1970	male	obesity
$t3$	1970	male	diabetes
$t4$	1981	male	diabetes
$t5$	1981	female	obesity
$t6$	1982	female	diabetes
$t7$	1982	female	cold

Table 3-2 Anonymized medical record generated from Table 3-1 ($k = 2$)

	Birth	Gender	Problem
$t1$	1970	male	cold
$t2$	1970	male	obesity
$t3$	1970	male	diabetes
$t4$	1981	human	diabetes
$t5$	1981	human	obesity
$t6$	1982	female	diabetes
$t7$	1982	female	cold

- Data providers can define an item as an open attribute. Open attributes will be published without any preservations of privacy if data users require publishing of the item. Open attributes are also able to be regarded as quasi-identifiers.

3.1.2 Request rule

Data users can request requirements using request rules as follows to obtain data from DTI4SFS. The data user creates the request rule according to the relevant publishing rule. Data users can know the detail of the provided data from regarding publishing rule.

- Request of an anonymization method and its level.
- Request of a combination of sensitive attributes.
- Request of quasi-identifiers. Open attributes are also candidates of the selection.
- Request of open attributes.

3.2 One-directional anonymization

As shown in Section 2.5, de-anonymization must be considered when publishing anonymized data. In other words, when multiple versions of anonymized data anonymized from the same data are published, an attacker who collected the multiple data may obtain another data that anonymity level is out of the permission of the publishing rule.

3 Data transaction infrastructure

Table 3-1 is an example of a medical record data table. Table 3-2 and Table 3-3 are anonymized data tables anonymized from Table 3-1 in different ways. Their anonymity levels are $k = 2$ and $k = 3$, respectively. It looks there are no problem when the data provider of the medical record permits $k \geq 2$. However, this situation actually causes de-anonymization when a malicious data user obtains both Table 3-2 and Table 3-3 because Table 3-1 that anonymity level is not permitted by the data user ($k = 1$) can be created by the two anonymized data tables. One cause of this problem is that previously published data is not referenced in the anonymization process; as a result, the coherence between the and data was severed. Table 3-4 is another example of an anonymized data table. This data table was generated by anonymizing Table 3-2 instead of anonymizing Table 3-1, to maintain coherency in masking and generalization. Publishing Table 3-4 instead of Table 3-3 avoids the problem described above since Table 3-1 cannot be obtained even if the malicious data user combines Table 3-2 with Table 3-4. The proposed policy of anonymizing process (one-directional anonymization) can prevent further leaks of privacy information. The proposed infrastructure (DTI4SFS) adopts one-directional anonymization to prevent de-anonymization caused by data sharing although the prevention is effective only for data published from a single DTI4SFS.

3.3 Design of DTI4SFS

DTI4SFS can be divided into four organizations as follows. Figure 3-2 represents proposed organizational structure and data connections between the organizations.

Table 3-3 Anonymized medical record generated from Table 3-1 (1) ($k = 3$)

	Birth	Gender	Problem
<i>t1</i>	19*	male	cold
<i>t2</i>	19*	male	obesity
<i>t3</i>	19*	male	diabetes
<i>t4</i>	19*	male	diabetes
<i>t5</i>	198*	female	obesity
<i>t6</i>	198*	female	diabetes
<i>t7</i>	198*	female	cold

Table 3-4 Anonymized medical record generated from Table 3-2 (2) ($k = 3$)

	Birth	Gender	Problem
<i>t1</i>	1970	male	cold
<i>t2</i>	1970	male	obesity
<i>t3</i>	1970	male	diabetes
<i>t4</i>	198*	human	diabetes
<i>t5</i>	198*	human	obesity
<i>t6</i>	198*	human	diabetes
<i>t7</i>	198*	human	cold

3 Data transaction infrastructure

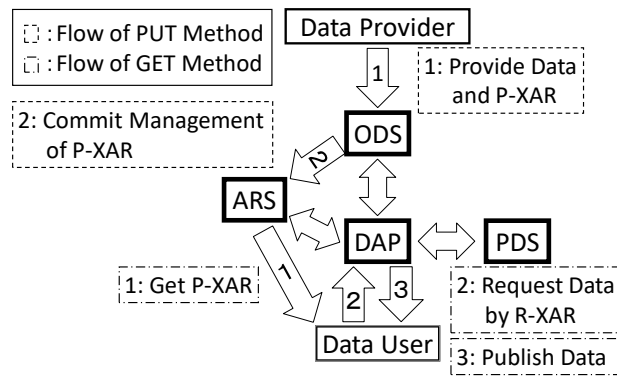


Figure 3-2 Design of the proposed infrastructure

3.3.1 Original data storeroom organization

Original Data Storeroom organization (ODS) manages data provided by data providers. When providing data to ODS, a data provider prepares data for publishing and provides a publishing rule by utilizing a specially designed format. This format is termed XML-based Anonymization Sheets (XAS). The details of XAS are described in Section 3.4. Publishing rule descriptions utilize a subset of XAS, termed XML-based Anonymization Rules (XAR). The data provider converts data provided to ODS to XAS format and provides the converted data as D-XAS with its publishing rule (P-XAR). D-XAS includes a link to the P-XAR to make a connection between the D-XAS and the P-XAR. ODS should be responsible for maintaining the original data written as D-XAS in a secure manner. This data registration process is based on the PUT method.

3.3.2 Anonymizing rules storeroom organization

Anonymizing Rules Storeroom organization (ARS) manages P-XAR. P-XAR will be openly published to data users. Since P-XAR includes an index of the relevant D-XAS, data users can know the which contents are included without reading D-XAS. Privacy concern would not be happened because P-XAR does not contain any private information. P-XARs stored in the ARS can exhibit data when it is available for its secondary use. A P-XAR is stored by utilizing a PUT method issued by ODS. Data users can generate their request rules (R-XARs) based on the relevant P-XAR.

3.3.3 Data anonymizing and publishing organization

Data Anonymizing and Publishing organization (DAP) anonymizes D-XAS according to its P-XAR and R-XAR. The R-XAR is generated and submitted to DAP by the data user in advance. The DAP receives the header of the requested D-XAS to access the link of the R-XAR. This header information does not include data. This header information is also described by using an XAR termed H-XAR; the DAP verifies its compliance by checking with the R-XAR and P-XAR requested from the ARS, according to the H-XAR. In this process, the data user utilizes a GET method in conjunction with the R-XAR option. If it returns a compliance error, the data user

3 Data transaction infrastructure

receives an appropriate error message based on the HTTP error message protocol. If no error occurs, DAP issues a GET message to obtain the D-XAS from the ODS, and issues a subsequent GET message to receive the published XAS (P-XAS) from Published Data Storeroom organization (PDS). PDS is described in the next Section (3.3.4). The DAP generates a P-XAS as anonymized data, and the response from the data user sent the R-XAR. The user receives the anonymized data resulting from the GET method. Finally, the DAP stores the generated P-XAS issues by utilizing the PUSH method. This P-XAS is utilized to prevent further privacy leaks.

3.3.4 Published data storeroom organization

PDS manages data previously published by the DAP as P-XASs. It may store all anonymized data generated by the DAP. However, to optimize data storage capacity, it is sufficient for the PDS to store only one P-XAS as anonymized data for each D-XAS, according to the one-directional anonymization policy. When generating P-XASs from D-XASs according to the requested R-XAR, it is sufficient to generate P-XASs according to the R-XAR, and store the P-XAS to the PDS. However, when generating another P-XAS from the same D-XAS according to another R-XAR, the DAP should obtain all P-XASs related to the D-XAS from the PDS. The DAP should consider all of these P-XASs when generating new P-XASs to observe P-XARs. The proposed infrastructure adopts the one-directional anonymization policy to avoid this process as following steps.

1. DAP generates P-XASs according to P-XARs, instead of R-XARs, and stores it in the PDS. Therefore, the PDS stores the anonymized data, and it is anonymized according to the declared level in P-XAR. This P-XAS is not sent to the users if the requested level in the R-XAR is higher than the level in the P-XAR; this indicates the value is larger than that of the P-XAR in k -anonymity.
2. DAP generates P-XASs according to the R-XARs. In this generation, the DAP only uses the initial P-XAS generated from the P-XAR in the previous step. DAP generalizes new P-XASs by generalizing appropriate values according to the R-XARs from the initial P-XAS. The DAP does not revert any of the generalized values in the first P-XAS. Therefore, a one-directional anonymization process should be considered.
3. DAP can generate any type of P-XAS that satisfies both the R-XAR and the P-XAR by following the process described in the first and the second steps. In a scenario where k -anonymity and l -diversity are mixed, it is sufficient to generate a P-XAS that has a lower anonymity level than k -anonymity and l -diversity. For example, assume that 3-anonymity and 3-diversity are permitted in P-XARs, and 4-diversity is requested by R-XAR. In this case, DAP generates the initial P-XAR by utilizing 3-anonymity. The DAP can generate any type of P-XAR by utilizing the initial P-XAR, according to the one-directional anonymization process.

To enable the data transfer between these organizations, data providers and data users will utilize Secure Sockets Layer (SSL) and Public Key Infrastructure (PKI) if they transfer the data over the Internet. In the following discussions, four organizations are exhibited to clarify each

3 Data transaction infrastructure

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <?xml-anonymize type="text/xas" href="p-xar.xas"?>
3 <list>
4 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:v="http://www.w3.org/2006/vcard/ns#">
5 <v:Kind rdf:about = "http://foo.com/me/hogehoge" >
6 <v:fn>Hoge Foo</v:fn>
7 <v:bday>1980-01-01</v:bday>
8 <v:hasTelephone>
9 <rdf:Description>
10 <rdf:value>+81-45-566-1454</rdf:value>
11 <rdf:type rdf:resource="http://www.w3.org/2006/vcard/ns#Work"/>
12 <rdf:type rdf:resource="http://www.w3.org/2006/vcard/ns#Voice"/>
13 </rdf:Description>
14 </v:hasTelephone>
15 <v:hasAddress>
16 <rdf:Description>
17 <v:street-address>123-45 Hoge Village</v:street-address>
18 <v:locality>FooCity</v:locality>
19 <v:postal-code>5555</v:postal-code>
20 <v:country-name>Japan</v:country-name>
21 </rdf:Description>
22 </v:hasAddress>
23 </v:Kind>
24 </rdf:RDF>
25 <OfficeScale>100ha</OfficeScale>
26 <PowerConsumption>10kWh</PowerConsumption>
27
28 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:v="http://www.w3.org/2006/vcard/ns#">
29 <v:Kind rdf:about = "http://foo.com/me/db" >
```

Figure 3-3 D-XAS example (extract)

role. It is possible to merge some of them into a single organization.

3.4 XML-based anonymization sheets

In this study, XML-based Anonymization Sheets (XAS) as a format to define the rules and data descriptions is proposed. XAS is designed with reference to Extensible Markup Language (XML) because XML is a popular data format in organizations. For example, IEEE1888, which is used for communication of IoT devices, adopts XML as the data format. To distinguish the rules from the data, XML-based Anonymization Rule (XAR) is also proposed as a subset of XAS. XAS and XAR differ because XAR does not contain data as contents. Transactions in DTI4SFS utilize the XAS and its subset, XAR. Figure 3-3 lists an example of D-XAS. It includes the information to enable anonymization, including combinations of the sensitive attribute names and quasi-identifiers, permitted anonymization methods and their levels, and data attributes such as created date, updated date and history, ownership, copyrights, comments, and others. Figure 3-4 lists an example of a P-XAR. It declares the required anonymization methods and their levels whereas it does not contain raw data. To enable generalizing processes such as masking, it can define the delimiter for distinguishing data sections. In this example, “BirthDay” is split using the ‘-’ character. During the anonymization process, the character is used to define the generalization boundary. If the data employs a general and standardized format, for example, BirthDay should

3 Data transaction infrastructure

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <anonymize>
3 <head>
4 <publishacceptance sensitive="divisional" quasi="divisional" />
5 <firstdatasetposition>
6 <list>
7 <rdf:RDF />
8 </list>
9 </firstdatasetposition>
10 <sensitive type="k(>=3), l(>=2)">
11 <rdf:RDF>
12 <v:Kind>
13 <v:hasTelephone>
14 <rdf:Description>
15 <rdf:type number="2" />
16 </rdf:Description>
17 </v:hasTelephone>
18 </v:Kind>
19 </rdf:RDF>
20 <PowerConsumption />
21 </sensitive>
22 <sensitive type="k(>=3), l(>=2)">
23 <OfficeScale />
24 </sensitive>
25 <group name="addr" type="quasi" level="k(>=3), l(>=3)" />
26 </head>
27 <rdf:RDF>
28 <v:Kind>
29 <v:fn note="Full Name" />
30 <v:bday note="BirthDay" type="quasi" level="k(>=2)" split="-" />
31 <v:hasTelephone>
32 <rdf:Description>
33 <rdf:value note="TelephoneNumber" type="open" split="\s" />
34 <rdf:type note="Number Type" attribute="rdf:resource" number="2" />
35 </rdf:Description>
36 </v:hasTelephone>
37 <v:hasAddress>
38 <rdf:Description note="Addresses">
39 <v:street-address group="addr" priority="4" />
40 <v:locality group="addr" priority="3" />
41 <v:postal-code group="addr" priority="2" />
42 <v:country-name group="addr" priority="1" />
43 </rdf:Description>
44 </v:hasAddress>
45 </v:Kind>
46 </rdf:RDF>
47 <OfficeScale note="OfficeScale" />
48 <PowerConsumption type="open" note="PowerConsumption" />
49 </anonymize>
```

Figure 3-4 P-XAR example

be separated by ‘-,’ it can generalize the data entry by referring to the default rule. As an additional feature, the data provider can publish data without any data publishing limits such as anonymization by selecting “open attribute,” which was described in Section 3.1.1. This feature allows use cases such as a data provider publishes data samples to publicize the data’s availability.

The secondary data user can request access to the open attributes by utilizing R-XAR. Figure 3-5 lists an example of an R-XAR. If the secondary data consumer requests attribute identified

3 Data transaction infrastructure

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <anonymize type="k(3)">
3 <head>
4 <sensitive>
5 <rdf:RDF>
6 <v:Kind>
7 <v:hasTelephone>
8 <rdf:Description>
9 <rdf:type number="2" />
10 </rdf:Description>
11 </v:hasTelephone>
12 </v:Kind>
13 </rdf:RDF>
14 <PowerConsumption />
15 </sensitive>
16 <group name="addr" type="quasi" />
17 </head>
18 <rdf:RDF>
19 <v:Kind>
20 <v:bday />
21 <v:hasTelephone>
22 <rdf:Description>
23 <rdf:value note="PhoneNumber" type="quasi" />
24 </rdf:Description>
25 </v:hasTelephone>
26 </v:Kind>
27 </rdf:RDF>
28 <PowerConsumption note="PowerConsumption" />
29 </anonymize>
```

Figure 3-5 R-XAR Example

as quasi-identifiers, DAP publishes anonymized data that contains attributes calculated as quasi-identifiers. The user also declares the required anonymization method, privacy protection level, sensitive attributes combinations, open attributes, and quasi-identifiers utilizing the R-XAR.

The formats of XAS and its subset XAR utilize the Cascading Style Sheets (CSS) format and the Semantic Web standard. The XAS can be processed utilizing an XML schema, RDL schema, OWL method, and other related tools.

3.5 Further studies for DTI4SFS

DTI4SFS enables data sharing while preserving included private information by using anonymization. The core process of DTI4SFS is anonymization, and this tends to be a bottleneck of the entire process flow because the anonymization process contains iterative sub-process of a large amount of data. In this study, a hardware implementation of the anonymization process is proposed to enhance transaction ability of DTI4SFS. Namely, the hardware implementation suppresses the bottleneck to transact a large amount of data at high speed. Conventional studies have a drawback of information loss due to the implementation cost. The proposed method overcomes the problem. Detail of the conventional studies are discussed in Section 4.1.

When focusing on the information value of shared data, change of the values in the data such

3 Data transaction infrastructure

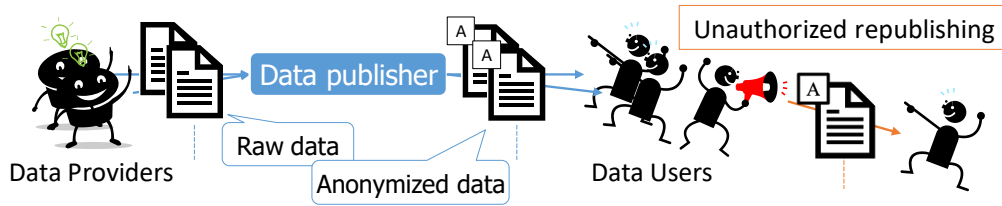


Figure 3-6 Unauthorized republishing in the proposed infrastructure

as time-series data is important for some data users rather than data itself. For instance, when supposing control of electric power consumption, the transition of the electricity usage is important for the applications such as peak shift. In this study, an anonymization method to share time-series data while keeping the information of the transition is proposed. Since this method generates several patterns of the datasets, the number of the datasets which transition patterns of the values are similar to each other fulfills k -anonymity. In addition, this method is effective for load distribution of anonymization. Electricity usage data can be anonymized at the location close to all of the relevant data sources due to the regionality. Conventional approaches and these drawbacks are discussed in Section 5.1. The discussion shows the anonymization approach was not well studied so far.

Another problem exists after data publishing to data users. DTI4SFS publishes according to the one-directional anonymization policy. Therefore, the same anonymized data are published to multiple data users who requested using the same R-XAR. The problem happens when one of the data users republishes the received data without any authorization (Figure 3-6). However, the malicious data user cannot be identified from the republished data since the published data are not unique each other. For solving this problem, in this study, a watermarking method for anonymized data is proposed. Watermarking is techniques to invisibly add information into data. If the proposed watermarking method adds information of the received data user, the malicious data user can be identified from the anonymized data republished by the malicious data user. The proposed method would be effective to suppress the unauthorized republishing. Conventional watermarking methods are not suitable to apply to anonymized data because anonymized data do not have particular order of the records whereas the conventional methods rely on order of records in the data. Additionally, modifications for the watermarking have to keep the anonymity of the data. Further discussion is shown at the beginning of Section 6.

4 Hardware implementation of anonymizer

DTI4SFS enables data sharing from data providers to data users while preserving private information in the data. As shown in Section 3.3, DTI4SFS can be divided into four organizations. Especially, DAP is the core organization in the proposed infrastructure.

All data provided by data providers are anonymized in DAP for privacy preservation. Anonymization generalizes values in given data so that all records in the data are not unique enough. On the other hand, the anonymization process tends to be a bottleneck of the entire transaction due to its high calculation cost compared with the other process. This is because the sub-processes of the anonymization such as generalization of values and confirmation of the current status of privacy metrics are iterated while referring entire related records. If the anonymization process becomes faster, the bottleneck would be overcome. Moreover, thanks to the high throughput, the anonymization processor would enable to anonymize data at line speed.

4.1 Existing studies and problem definition

There have been numerous studies on anonymization models and software implementations of anonymizers, such as automatically detecting fields to anonymize and anonymizing time-series data [45] [46]. Contrastingly, the number of this kind of studies is very little when it comes to hardware implementation. There are some studies on utilizing Field Programmable Gate Arrays (FPGAs) for data mining. However, they do not consider the anonymity of the mined data [47] [48].

Sawada et al. proposed an FPGA-based anonymizer that utilizes Ternary Content Addressable Memory (TCAM) and caches to anonymize network data streams [49] [50]. TCAM is one of CAMs, which are special types of memory. General memories called Random Access Memory (RAM) return a stored data entry according to given address of the memory. In contrast to the RAM, a CAM returns a list of data entries which are matched to a given keyword. Especially, TCAM allows '0', '1', and '-' (don't care) to express data entries whereas data entries of general CAMs are expressed by '0' and '1' only. Therefore, TCAM is suitable for anonymization using masking by using '-' as a mask of anonymization. Additionally, TCAM enables single-cycle data table lookup. Hence, it is an ideal memory for anonymizers, which require frequent data lookups. The study of Sawada et al. showed enough throughput for high-speed networks over 1GbE, but its largest disadvantage is the use of TCAM memory.

The disadvantage of using TCAM is its high circuit implementation cost. When TCAMs are used for anonymizers on FPGAs, their processing window size becomes smaller because of the large circuit utilization. Since anonymization is performed within the processing window, larger window sizes will reduce the information loss ratio during anonymization. Therefore, TCAM-

4 Hardware implementation of anonymizer

based anonymizers tend to have higher information loss due to the limit of window size.

This section focuses on addressing this problem by proposing an anonymizer that has a large window size and a small enough circuit size for implementation on a mid-range FPGA device. Being implementable on a mid-range FPGA brings benefits, such as cost-effectiveness and availability. These benefits are important when considering implementation cost of DAP because if the implementation cost becomes lower, the anonymization process can be distributed to produce further high-throughput anonymization. For the distribution, allocating the proposed anonymizer on a network device is appropriate because it is rational that data provided by data providers are anonymized while the data are sent through the network in DTI4SFS. Additionally, it is useful for data providers who want to anonymize data before providing the data to DTI4SFS. This pre-anonymization would reduce load of anonymization process at DTI4SFS. For these reasons, this study supposes that the proposed anonymizer is implemented on a network device. For the use case of the proposed anonymizer, an anonymization of LAN traffics such as public network at event sites and public facilities while capturing the traffics at the gateway of the LAN is supposed. In the use case, the network traffics can be analyzed to survey Internet usage. The required throughput in the networks is generally 10Gbps or at least 1Gbps. The proposed anonymizer aims to overcome the throughput.

4.2 Proposed architecture

As mentioned in Section 4.1, most existing studies on hardware-based anonymizers rely on TCAM and thus result in high implementation costs. For example, the maximum window size possible on a Virtex-5 XC5VLX330T FPGA device is merely 256 despite the fact that the FPGA has the largest circuit area in its class. Implementing the TCAM-based architecture on dedicated Application Specific Integrated Circuit (ASIC) circuits will allow larger window size and improve the performance. However, the cost for producing such circuits for low-volume products, such as internet routers, will increase drastically.

Given these limitations, it is ideal to implement the anonymizer on an FPGA-device to lower the cost and allow flexibility. For maintaining the information loss ratio at a low level and to make the circuit small enough for FPGAs, in this study, a RAM-based anonymizer is proposed. The use of RAM will lower the circuit utilization and, therefore, increase the window size and lower the information loss ratio.

4.2.1 Overview of the proposed architecture

The architecture of the proposed anonymizer is illustrated in Figure 4-1. The main components of the anonymizer are the controller, data generator, RAM, buffer, Bloom filter, and hash modules. The controller manages anonymization process while the data generator generates values to update anonymization status. The generated values are sent to the RAM via the buffer. The buffer is also used to store a tuple in the processing window one by one. The use of RAM is what distinguishes this anonymizer from existing ones. Hashing is used for addressing the memory to

4 Hardware implementation of anonymizer

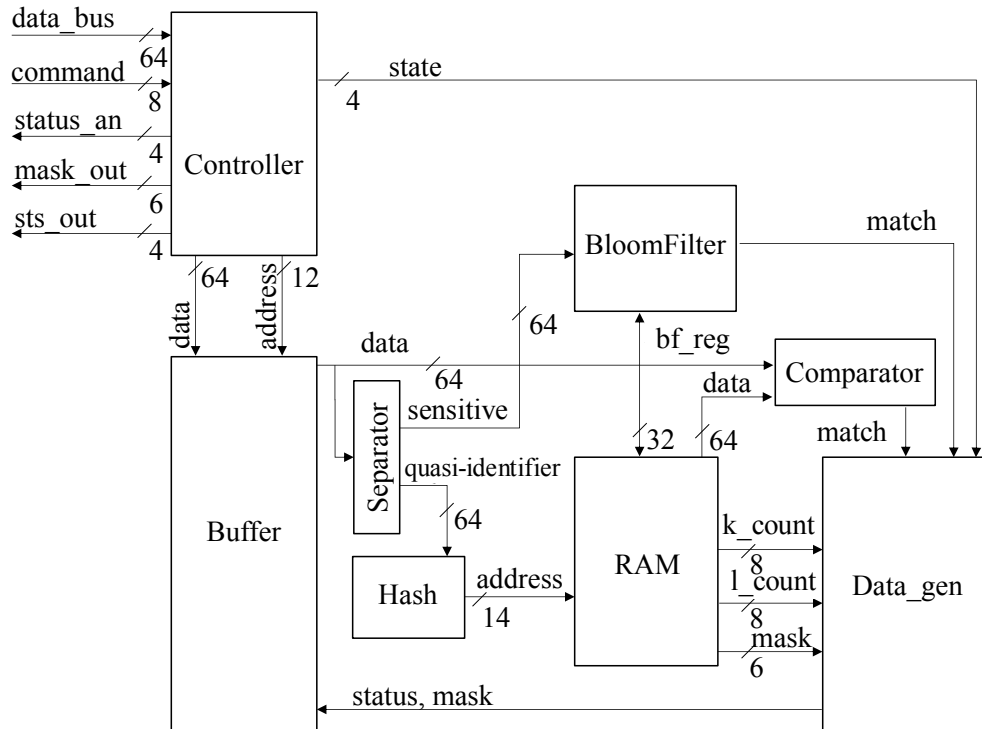


Figure 4-1 Architecture of the proposed anonymizer

enable fast lookups using the RAM. The Bloom filter is used to check whether each q^* -block fulfills required l -diversity.

The proposed anonymizer adopts masking described in Section 2.1.4 as the generalization technique. IP address was set to the quasi-identifier for the evaluation to compare with the implementation using TCAM.

4.2.2 RAM and buffer

The RAM and buffer memory used for the anonymizer is a distributed RAM, which is directly implemented on the FPGA logic. Distributed RAM was chosen for this anonymizer for its asynchronous read capability. Asynchronous read allows for faster table lookups and, therefore, improve the overall throughput of the anonymizer.

The data fields of the RAM and buffer memory are shown in Table 4-1 and Table 4-2, respectively. The “data” field holds the original data that was transferred to the anonymizer for the anonymization process. This 64-bit field holds both the quasi-identifier and the sensitive attribute. The allocation of the quasi-identifier and the sensitive attribute can be adjusted by sending commands to the anonymizer, which will be explained in Section 4.2.3. The “ k_count ” and “ l_count ” fields hold the anonymity levels of the corresponding tuple. The “mask” field holds the number of bits of the quasi-identifiers that have been generalized. Table 4-3 explains examples of the masking. The “SET_SEP” in this table is the command to set the allocation of the quasi-identifier and the sensitive attribute. The value of SET_SEP represents the number of

4 Hardware implementation of anonymizer

Table 4-1 Data fields of RAM

Bits	Field
[123:120]	parity
[119:56]	quasi-identifier
[55:54]	status
[53:48]	mask
[47:40]	k_count
[39:32]	l_count
[31:0]	bloom filter register (bf_reg)

Table 4-2 Data fields of buffer

Bits	Field
[71:8]	data
[7:6]	status
[5:0]	mask

Table 4-3 Example of masking in the proposed anonymizer

SET_SEP	mask	data	quasi-identifier
2	1	00110000	00110***
2	3	00110000	001*****
4	2	00110000	00*****

bits of the part of the sensitive attribute from the rightmost bit in the data field. The mask field represents the number of masked bits from the rightmost bit of the quasi-identifier part. Therefore, the “quasi-identifier” field is the masked version of the data field masked from the rightmost bit that a number of the masked bits is the sum of the SET_SEP and mask fields. The “status” and “parity” bits hold the current status of the anonymization process. The status field represents whether the anonymization process is finished or not. The parity field represents whether the data entry is valid or not. The use of the parity field omits initialization process of the RAM at every cycle of anonymization.

4.2.3 Controller

The controller is a state machine to manage the anonymization process of the anonymizer. The controller state serves as the control signal for the other modules of the anonymizer. Other modules in the anonymizer, such as the data_generator and Bloom filter, change their function according to the status of the controller.

The controller also serves as the access controller for interfacing the anonymizer from external components. It receives the dataset to be anonymized and sends back the mask data as output. It also receives the command signals and adjusts the window size, k -anonymity, l -diversity, and separation parameters. The list and description of the commands are shown in Table 4-4.

Requirements of k -anonymity and l -diversity are set by the “SET_K” and “SET_L” command. The SET_SEP command is used to define the allocation of the quasi-identifier and sensitive attribute as described in Section 4.2.2. The “SET_W” command is used to define the processing

4 Hardware implementation of anonymizer

Table 4-4 Controller commands

Value	Name	Description
0x00	IDEL	Idle State. Do nothing.
0x01	SET_K	Set the k -anonymity target.
0x02	SET_L	Set the l -diversity target.
0x03	SET_SEP	Set the separating point of quasi-identifier and sensitive attribute on the <code>data_bus</code> .
0x04	SET_W	Set the window size.
0x05	DATA_S	Start of dataset.
0x06	DATA	Input data.
0x07	DATA_E	End of dataset.

Table 4-5 Values of `status_an` bus and their descriptions

Value	Name	Field
0x0	RSV	Reserved.
0x1	READY	Ready state. Ready for commands.
0x2	BUSY	Busy state. Do not send additional commands.
0x3	ERROR	Error.

Table 4-6 Values of `sts_out` bus and their descriptions

Value	Name	Field
0x0	IDLE	Idle state. No data output.
0x1	DATA_S	Beginning of output data.
0x2	DATA	Output data.
0x3	DATA_E	End of output data.

window size. The “DATA” command is used to send dataset to be anonymized. The start and end of importing the data is controlled by the “DATA_S” and “DATA_E” commands, respectively.

The controller sends status information of the anonymizer to the external components. The “`status_an`” and “`sts_out`” signals in Figure 4-1 exports status information of the anonymizer and data output, respectively. Table 4-5 and Table 4-6 show further information of the values sent from the two signals.

4.2.4 Data generator

The `data_generator` composes the signal generated by the RAM, buffer, and Bloom filter modules and generates a new entry for the RAM data table. The `data_generator` analyzes the contents of the RAM, buffer, and Bloom filter outputs and perform the necessary operation for each state of the anonymization process. The operation includes calculations of values to update the `k_count`, `l_count`, `mask`, and `status` fields. Details of this data maneuver are explained in Section 4.3.

4 Hardware implementation of anonymizer

4.2.5 Hash

The anonymizer uses a technique called hashing for addressing the memory for data lookup. Hashing generates a hash value from the data, and the hash value is used as the address of the memory where the data will be stored. This hashing function enables very quick table lookups since hash value calculation is the only necessary operation.

While the use of hashing for data lookup brings many benefits, including faster operation and lower circuit utilization, a hash collision cannot be avoided. Yamaguchi et al. [51] evaluated the hash collision rates and the hardware performances of different hash functions under network workloads. According to the evaluation in the study, a hash collision rate depends on the property of data to be hashed. Therefore, it is difficult to identify the best hash function in terms of hash collision rate. Cyclic Redundancy Check (CRC) [52] was selected as the hash function for the proposed architecture since it has the smallest delay.

The hash function used for the proposed design is a CRC32 with a polynomial 4-1. Of the 32-bit hash value, n lower significant bits are used as the memory address. n is the number of binary digits necessary to address the entire processing window. For example, for a window size of 4096, $n = \log_2 4096 = 12$.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

4-1

4.2.6 Bloom filter

The use of the Bloom filter also makes the proposed design unique over previous designs. The Bloom filter is a data structure that allows fast query of the members of a dataset [53]. The Bloom filter is appropriate for the proposed anonymizer in two aspects. One of the aspects is implementation cost. The Bloom filter consists of bit arrays and hash functions only. Hence, the Bloom filter requires little implementation cost. The other aspect is a small delay. The Bloom filter processes at low latency because the Bloom filter registers and refers data by hashing and logical operations only. The bloom filter runs at a single cycle in the hardware implementation.

The anonymizer adopts the Bloom filter to verify whether the tuples in the data have met l -diversity. To verify l -diversity, how many types of sensitive attributes exist in each q^* -block must be counted. When a new tuple is added to a q^* -block, the Bloom filter is used to check whether the sensitive attribute in the new tuple already exist in the q^* -block or not. The Bloom filter returns true when the sensitive attribute exists. The relevant l_{count} field is counted up when the Bloom filter returns false.

The Bloom filter requires two hash functions. In this study, CRC32 with the polynomial 4-1 with different initial values are used for both of the two functions. The “bloom filter register (bf_reg)” field in the RAM holds the 32-bit string for the Bloom filter. The bf_reg contains the sensitive attribute of the q^* -block.

Using the Bloom filter rarely makes false positive because the Bloom filter uses hash functions.

4 Hardware implementation of anonymizer

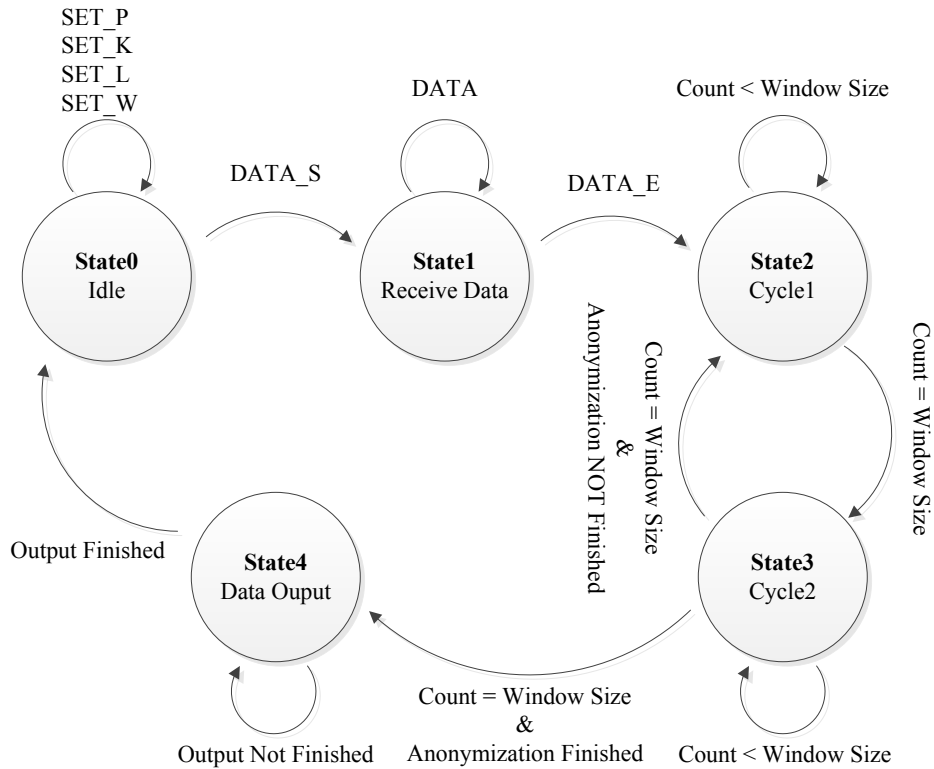


Figure 4-2 State diagram of the controller

When the false positive happens, the Bloom filter returns true even though the sensitive attribute does not exist in the q^* -block. Fortunately, the false positive is acceptable regarding privacy preservation because the false positive promotes further anonymization. Although the false positive increases information loss, the anonymizer adopts the Bloom filter to reduce the circuit implementation cost whereas the false positive is allowed.

4.3 Details of the anonymization process

The fundamental functionality of the proposed anonymizer is to receive datasets that require anonymization and determine the mask length of the quasi-identifier field for each tuple in the dataset. The entire process can be divided into the following procedures: command and data reception, anonymization, and data output.

In the command and data reception state, the anonymizer accepts commands and data from external devices. Next, the anonymization process is continued until all tuples in the dataset achieve the requested anonymity levels. When the anonymization process finishes, the mask bits are sent out from the output bus.

4.3.1 State transition

The state diagram of the controller is illustrated in Figure 4-2. There is a total of 5 states in the controller. State0 and 1 correspond to the command and data reception, state2, and 3

4 Hardware implementation of anonymizer

anonymizations, and state4 data output.

State0 is the initial state of the anonymizer, and servers as the idle point until a new anonymization task is started. When there are incoming commands, the necessary data maneuvers are performed accordingly. When the “DATA_S” command, which marks the beginning of a new dataset, is received, the controller moves its state to state1. The incoming data is stored in the Buffer until the controller receives the “DATA_E” command. This command moves the state to state2.

State2 and 3 are the two states in which the actual anonymization is performed. The anonymization process is divided into two cycles: cycle1 and cycle2. The procedures of each cycle are repeated until it reaches the end of the processing window, and in cycle1, the state transit to cycle2. In cycle2, the state continues to state4 if all tuples have achieved anonymity, and if not, the state moves back to state2 (cycle1).

State4 sends the mask bits from the output bus. When the mask data for the entire dataset is sent out, the state moves back to state0 and waits for commands and new data.

4.3.2 Anonymization process

As mentioned in Section 4.3.1, the anonymization process is divided into two steps: cycle1 and cycle2. Pseudo code for the anonymization process is given in Figure 4-3.

4.3.2.1 Anonymization Cycle1

The first process of cycle1 is to determine if a tuple with the same quasi-identifier has appeared earlier. Every tuple in the buffer is sequentially read, and a hash value is generated for RAM table lookup. Since this value is used as the address of the RAM, tuples with the same quasi-identifier correspond to the same memory address. Therefore, each entry in the RAM represents a q^* -block. Therefore, if the RAM entry for the tuple has a valid entry, this means that the tuple with the same quasi-identifier has appeared at least once within the same processing window. This is true if the hash collisions do not occur. Unfortunately, the hash collision does occur. For the proposed architecture, hash collision is detected by comparing the quasi-identifier entry of both RAM and buffer. If both entries match each other, the RAM entry is valid, and the k_count field is incremented to show that this q^* -block has one more member. If the values do not match, this indicates a hash collision, and the quasi-identifier of both tuples are masked entirely to preserve data privacy.

The next process of cycle1 is to determine the l -diversity level. The l -diversity level is increased if the sensitive attribute of the tuple has not appeared in the same q^* -block. As mentioned in Section 4.2.6, the Bloom filter is utilized to test the existence of a tuple with the same sensitive attribute within the q^* -block. Every RAM entry, i.e. q^* -block, holds a 32bit bf_reg field, which is used as the Bloom filter register. Each time a sensitive attribute is tested, the bf_reg is read by the Bloom filter and tested. If the membership is true, this indicates that a tuple with the same attribute has appeared previously, and the l -diversity level stays the same. If false, there are no tuple within the q^* -block that has the same sensitive attribute, and the l -

4 Hardware implementation of anonymizer

```
while anonymization not finished do
  case cycle 1
    for each tuple in window do
      if new tuple then
        add sensitive attribute of tuple to bloomfilter
        k_count ← 1
        l_count ← 1
      end if
      else if not new tuple then
        k_count ++
        if bloomfilter member test = false then
          l_count ++
        end if
      end else if
    end for
    goto cycle2
  end case
  case cycle 2
    for each tuple in window do
      if k_count ≥ k_target && l_count ≥ l_target
        state = finished
      end if
      else
        mask ++
      end else
    end for
    if all state = finished then
      exit anonymization loop
    end if
    goto cycle1
  end case
end while
```

Figure 4-3 Pseudo code for the anonymization process

diversity level is advanced by incrementing the *l_count* value.

4.3.2.2 Anonymization cycle2

Cycle2 determines the anonymity level of each tuple and performs the necessary operations. The anonymity level of each tuple is stored in the *k_count* and *l_count* fields of the RAM entry corresponding to each tuple's *q**-block. If both *k*-anonymity and *l*-diversity levels reach the requested anonymity level, the tuple's status entry in the buffer is updated to the finished status. If the anonymity levels are not met, the mask bit is incremented and stored into the buffer. Tuples with mask value, which has reached the value that would mask the entire quasi-identifier, is also marked as finished. As these tasks are performed, the anonymizer checks whether all tuples have

4 Hardware implementation of anonymizer

achieved anonymity by placing a flag bit which turns low once a non-finished tuple appears; when all tuples meet anonymity requirements, the flag bit remains high. If the flag bit remains high at the end of cycle², the anonymizer exits the anonymization loop to proceed to the data output.

4.4 Evaluation of the anonymizer

The performance of the anonymizer was measured by the following three metrics: information loss, throughput, and circuit size. The information loss ratio is measured to determine the data retention rate after anonymization. Throughput evaluations show how fast the anonymizer is capable of anonymizing datasets. Lastly, the circuit size is evaluated to compare the implementation cost of the two designs.

For a fair comparison of the proposed anonymizer with the TCAM-based anonymizer, the proposed architecture was modified to make the two designs have the same functionality. The following are the modifications applied to the proposed architecture.

- Data input to the anonymizer regarding both quasi-identifier and sensitive attribute was fixed to 32 bits, respectively. The capability of modifying the assignment of quasi-identifier and sensitive attribute on the data bus does not exist on the TCAM architecture. Therefore, this functionality was omitted from the proposed architecture for evaluation.
- The command receiver functionality to set window size and anonymity parameters was removed.
- Data fields of RAM that are no longer necessary as a result of the design changes above was modified. The parity field was deleted, and the quasi-identifier field was shrunk to 32 bits.

4.4.1 Information loss ratio

The information loss ratio was measured to analyze how much data is retained after an anonymization process. Lower information loss ratio retains more data. Therefore, reducing information loss ratio would be more beneficial for the data users. To evaluate the anonymizer for network application usage, the workload used for the analysis was the source IP address, and the URL of HTTP GET requests of a laboratory office network. The network consists of approximately 100 clients, and the workload contains a total of around 20,000 requests. A software-based simulator for the RAM and TCAM anonymizers was used for this evaluation.

Figure 4-4 and Figure 4-5 show the relationship between information loss ratio and the window size of four patterns of l -diversities regarding TCAM and RAM anonymizers, respectively. According to the two figures, the information loss ratio decreases when the window size becomes larger. This is because large window size provides a large number of candidates of tuples to create a q^* -block without masking. The two figures also represent the relationship between the required anonymity level and the information loss ratio. The information loss ratio becomes larger when

4 Hardware implementation of anonymizer

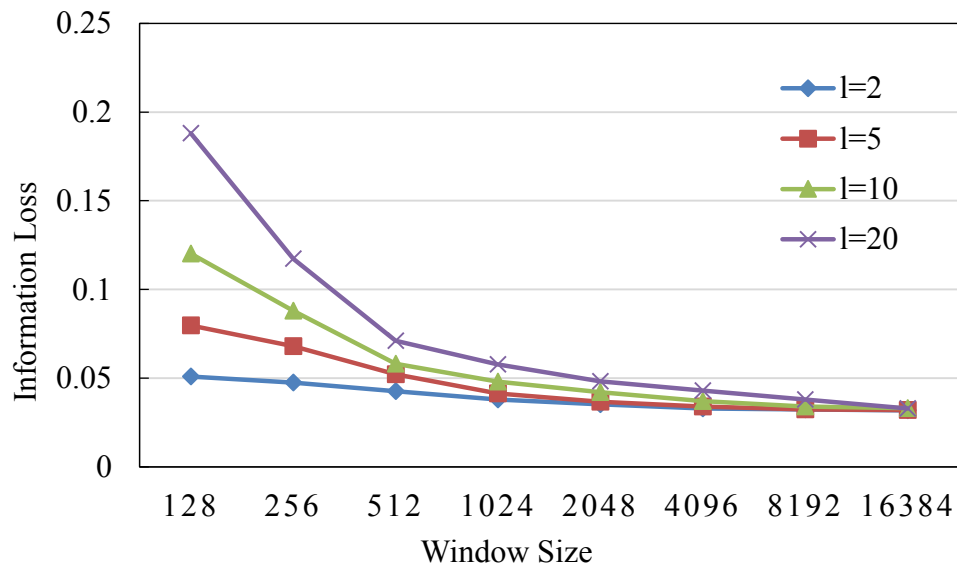


Figure 4-4 Relationship between information loss ratio and the window size of four patterns of l -diversities (TCAM)

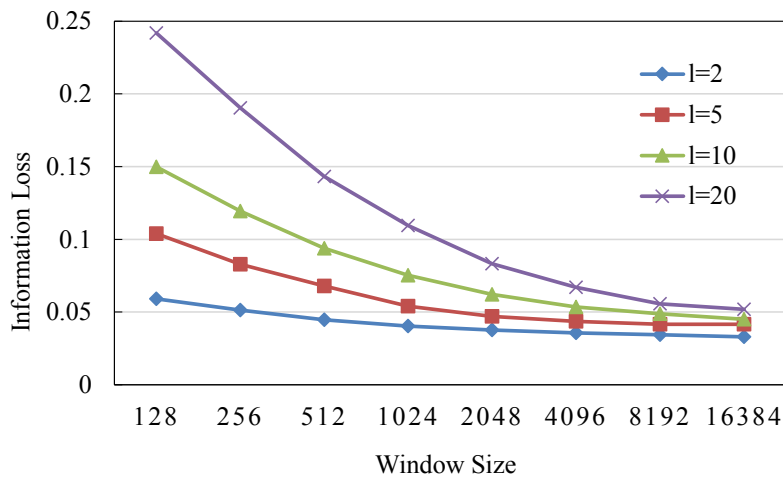


Figure 4-5 Relationship between information loss ratio and the window size of four patterns of l -diversities (RAM)

the large value of anonymity level is required.

Another type of comparisons regarding the information loss ratio of the TCAM and RAM anonymizers are given in Figure 4-6 to Figure 4-9. The two architectures are compared at different k -anonymity and l -diversity requirements. The dotted line shows the information loss ratio of the TCAM anonymizer when the window size is 256. This value is important since 256 is the maximum window size that Virtes-5 XC5VLX330T can accommodate.

From the results, it is clear that TCAM architecture has lower information loss ratios in any anonymity level when the window size is the same compared with the RAM architecture. The

4 Hardware implementation of anonymizer

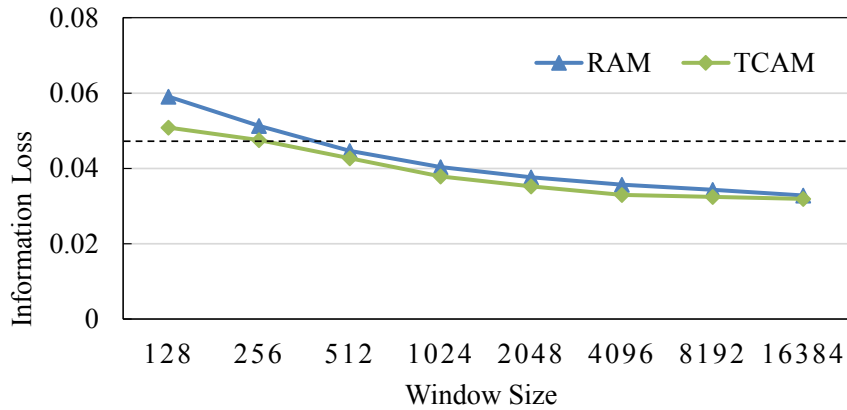


Figure 4-6 Information loss ratio ($k = 2, l = 2$)

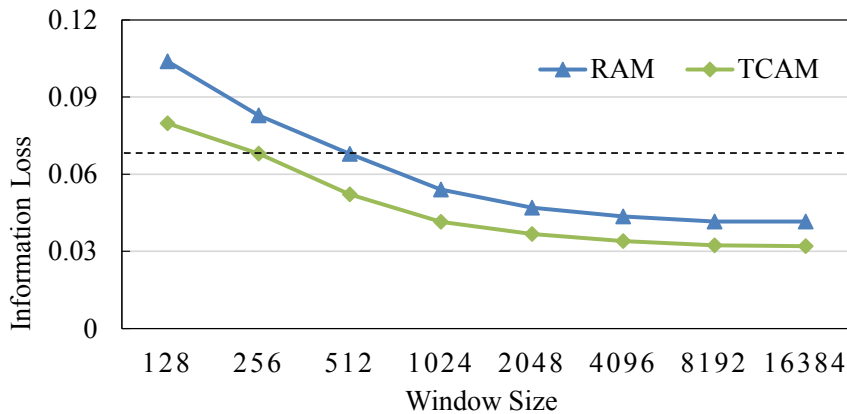


Figure 4-7 Information loss ratio ($k = 5, l = 5$)

RAM architecture has a larger information loss ratio for two reasons. The first reason is a hash collision. When a hash collision occurs, the entire quasi-identifier is masked to prevent anonymity shortfalls. This masking is the result of using hashing for table lookup and was not an issue for TCAM anonymizers.

The second reason is the false-positive error caused by the Bloom filter. A false-positive error of the Bloom filter results in the anonymizer determining that the l -diversity level is lower than actual; a positive result from the Bloom filter implies that the tuple with the same sensitive attribute exist within the q^* -block and that the l -diversity level is not increased. This false-positive error is not a problem for the anonymity level but has effects on the information loss ratio since the l -diversity level is determined to be lower than the actual and the mask bits are increased more than necessary.

Although the RAM architecture has a larger information loss ratio when the window size is the same, the ratio decreases enough that the levels go under those of TCAM at 256. When the window size was 1,024 or larger, evaluations under all anonymity levels showed less information loss than TCAM at 256. Since the RAM has a significantly less circuit utilization, the window size can be further extended when embedded on the same FPGA device. Further analysis of the circuit

4 Hardware implementation of anonymizer

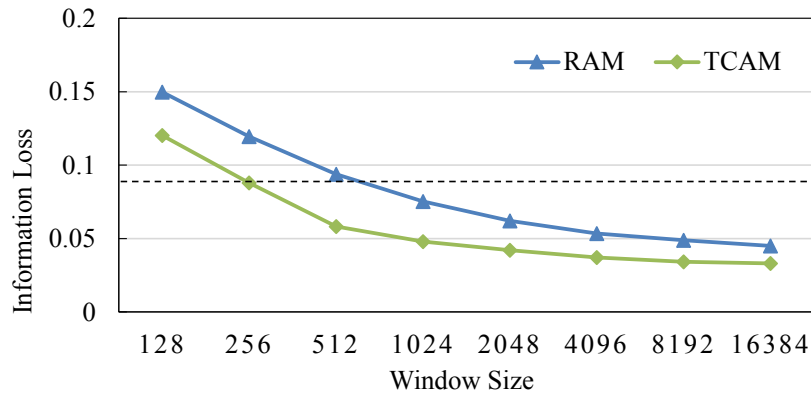


Figure 4-8 Information loss ratio ($k = 10, l = 10$)

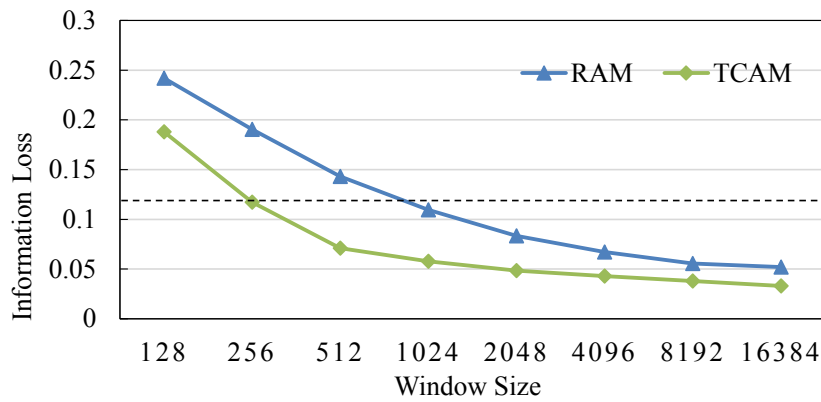


Figure 4-9 Information loss ratio ($k = 20, l = 20$)

size evaluation is shown in Section 4.4.2.

Another characteristic is that at window sizes over 4,096, the decrease in the information is very small. For example, the information loss ratio was reduced only by only 1.5% from window size of 4,096 to 16,384.

4.4.2 Circuit size

The circuit size evaluation was performed to analyze the circuit size of the two designs. Smaller circuit size is beneficial, as it will lower implementation costs. Circuit size was evaluated by comparing the resource utilization rates when implemented on an FPGA device. The target FPGA device was Xilinx Virtex-5 XC5VLX330T. Cadence NC-Verilog and Simvision 06.20 were used for circuit simulation, and ISE Design Suite 14.2 was used for synthesis and implementation.

The window size of the TCAM-based anonymizer was set at 256 tuples since it is the largest size that the FPGA can accommodate. For the RAM-based anonymizer, the circuit size evaluation was done with a window size of 256, 512, 1,024, 2,048, and 4,096. Table 4-7 shows the FPGA device utilization summary.

4 Hardware implementation of anonymizer

The device utilization of Table 4-7 shows that the RAM architecture has lower circuit utilization under any window size configuration. Results in Section 4.4.1 have shown that the RAM architecture achieves lower information loss rates than the TCAM architecture when the window size is larger than 1,024. When the slice LUT utilization of the RAM and TCAM architectures with the same configurations are compared, the RAM architecture had an 82.3% reduction. Even when the window size was at 4,096, RAM architecture had 39.5% lower utilization.

4.4.3 Throughput

The throughput of the anonymizer was evaluated with two metrics that are maximum frequency and tuples/second. The maximum frequency of the circuit shows how fast the circuit is capable of being operated. The device setup used for this evaluation is the same as those for circuit size evaluation. The maximum frequencies of the RAM and TCAM designs are 105 MHz and 70 MHz, respectively.

Although the RAM architecture has a higher operating frequency than the TCAM architecture, it does not necessarily mean that the RAM architecture has a higher throughput. The two anonymizers have different anonymization procedures, and the necessary clock cycle for completion is different. An original simulator was used to evaluate the actual throughput using real-life traffic. The simulator emulates the anonymization sequence of the TCAM and RAM-based anonymizers and measures the necessary clock counts for the entire anonymization process. Network traces with 20,000 HTTP requests were used as the workload. After the clock counts were obtained, the number of tuples each anonymizer is capable of handling per second at its maximum frequency was calculated.

The throughput of a software anonymizer was also evaluated for comparison. The evaluation was done on a Linux server with dual 2.9 GHz Intel Xeon E5-2690 processor and 128 GB DDR3 memory.

Table 4-7 Circuit utilization

	Number of Slice Registers		Number of Slice LUTs	
	(out of 207360)	(out of 207360)	Number used as Logic (out of 207360)	Number used as Memory (out of 54720)
TCAM (256)	3733 (0%)	42550 (20%)	9526 (4%)	33024 (60%)
RAM (256)	129 (0%)	2989 (1%)	1597 (0%)	1392 (2%)
RAM (512)	130 (0%)	4482 (2%)	1698 (0%)	2784 (5%)
RAM (1024)	128 (0%)	7528 (3%)	1960 (0%)	5568 (10%)
RAM (2048)	128 (0%)	13546 (6%)	2410 (1%)	11136 (20%)
RAM (4096)	128 (0%)	25739 (12%)	3467 (1%)	22272 (40%)

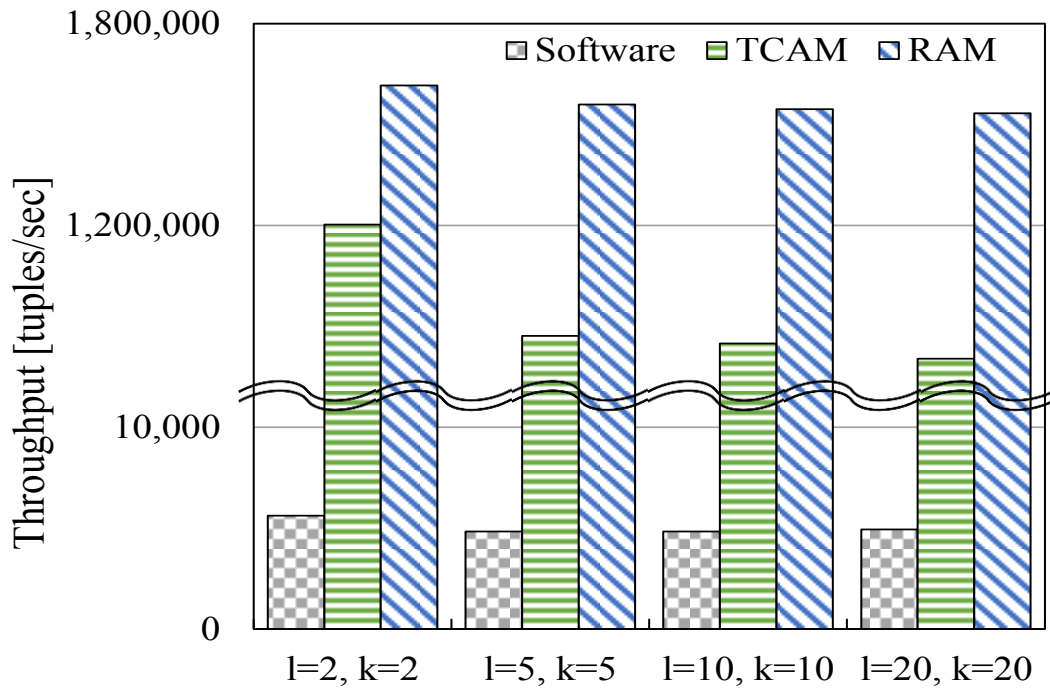


Figure 4-10 Throughput of the anonymizers

Figure 4-10 shows the results of the evaluation. The RAM architecture has a maximum throughput of 1.61M tuples/sec and had a 34% to 90% increase in throughput over the TCAM architecture, and when compared to the software implementation, the throughput improvement was 380 to 440x. Compared to the software anonymizer, parallelization of hash calculation, Bloom filter query/add, and data maneuvers contributed to increasing the throughput of hardware anonymizers.

When considering an application where portions of the network traffic are anonymized, the anonymizer does not have to anonymize the entire network workload. For example, in case where a single set of quasi-identifier and sensitive attribute (e.g. IP address and URL) is the subject for anonymization for every packet in a network with an average packet size of 1,000 bytes, the RAM architecture will be able to anonymize the IP and URL of network with bandwidth of up to 1.61M (tuples/sec) \times 1,000(bytes/packet) \times 8 = 12.9Gbps. This is enough throughput to anonymize contents of 10GbE and OC192 speed networks on the fly whereas the throughput of the software implementation is less than 1Gbps.

4.5 FPGA implementation

Based on the results of the evaluation, a RAM architecture with a window size of 4,096 was chosen for the final implementation on FPGA. The FPGA used for the implementation was Xilinx KC705 Kintex-7 evaluation board with XC7K325T-2FFG900C device onboard (Figure 4-11). Vivado 2014.2 was used for synthesis and implementation. In addition to the proposed anonymizer,

4 Hardware implementation of anonymizer

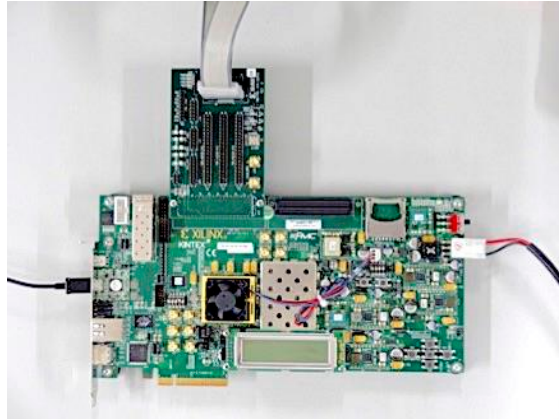


Figure 4-11 Xilinx KC705 Kintex-7 evaluation board with FPGA

Table 4-8 Implementation results

Number of Slice Registers	124 (0%)
Number of Slice LUTs	39973 (19%)
Number used as Logic	13861 (7%)
Number used as Memory	26112 (41%)
Maximum Frequency	71.3 MHz

the inserter module was implemented into the FPGA to insert data to the anonymizer for testing. XM105 debug card, and a logic analyzer was used to observe the output. The circuit size and the maximum operating frequency of the implemented anonymizer are given in Table 4-8.

Circuit utilization and maximum frequency for the implemented design is inferior to those of Section 4.4. This is the result of stripping off some functionality for evaluation. This increase in circuit utilization and decreased maximum frequency would have been evident if the additional functionality were added to the TCAM architecture. The implemented design has a maximum throughput of 8.75Gbps, which is 68% of the throughput calculated for the result in Section 4.4.3. The throughput of the implemented design is enough to anonymize 1GbE networks that are the supposed use cases mentioned in Section 4.1 whereas the throughput of the software implementation is not enough. Although the throughput of the implemented design is less than 10Gbps, the proposed anonymizer would overcome the throughput when the anonymizer is implemented on either a more high-performance FPGA or an ASIC. Additionally, the throughput of the implemented design is a little short of OC192 but enough for OC48.

4.6 Summary

In this section, a RAM-based anonymizer was proposed and implemented on an FPGA for data anonymization for DTI4SFS. Hashing and Bloom filters were utilized to effectively anonymize datasets while retaining throughput and lowering the circuit utilization cost. The proposed anonymizer showed 82.3% reduction in circuit size compared to existing anonymizer based on TCAM. The maximum throughput of the anonymizer was 8.75Gbps, which is enough for 1GbE and multiple OC48 speed networks.

5 Anonymization method to share electricity usage data

As described in Section 2.1.4, anonymization generally modifies values in data until the data does not contain unique tuples. However, this kind of anonymization methods is sometimes not appropriate for some data types. Time-series data such as sensor data are categorized to the data types.

A time-series data table represents a change of the values, i.e. data transition. The tuples in the time-series data table consist of a combination of a timestamp and values related to the time. Therefore, data users utilize time-series data to know the information about how the values change rather than the information about values themselves in each tuple. Accordingly, the general anonymization methods described in the previous paragraph do not suit the time-series data. For instance, when anonymizing the time-series data using the general anonymization method, timestamps could be generalized to create q^* -blocks. This anonymization would spoil the information of the rapid change of values. Here is another example of the situation that the q^* -block consists of tuples of the same timestamp by mixing multiple time-series data tables. This example would also ruin the information of the transition. There is another concern from the aspect of privacy preservation. The concern is that a q^* -block that consists of tuples related to a single individual could be generated. Such anonymized data still contains unique information.

These problems can be solved if the single time-series data table is regarded as the unit of anonymization like a tuple in a data table in the general anonymization methods. In other words, k -anonymity is achieved when time-series data tables are classified to some groups while each of the groups has k of the time-series data tables at least. The classification result shows the information of the change of the values for each of the groups. In addition, the result does not contain any group related to a single individual only. If such anonymization method exists, the proposed infrastructure can anonymize time-series data while keeping the data quality required by the data users.

This study proposes an anonymization method to share time-series data while supposing data sharing in smart cities. Smart city is a city that effectively controls infrastructures related to cities to provide convenient and comfortable life. The smart city has a smart grid that is one of the electric grids, which provides electricity from power suppliers to consumers. Smart grids are different to conventional electric grids because smart grids allow intercommunication between the suppliers and the consumers. Smart meters are used in smart grids instead of conventional power meters for the intercommunication. Smart meters are installed as essential devices for smart houses and home energy management systems. It allows power suppliers to acquire the electricity usage of the target household as electricity usage data. Analysis of the acquired data renders smarter services from power suppliers. For instance, the total amount of used electricity

5 Anonymization method to share electricity usage data

during peak times would be reduced by shifting the peak electricity usage in each household. The technique to adjust the electricity bill for electricity usage control is referred to as a demand response (DR). The DR service is used to cut or shift the peak electricity usage to maintain the demand–supply balance at a period. However, when the company under contract only captures the smart meter data, it becomes difficult to manage the fine DR service in the target region. This is because all power companies in the region cannot share the usage data for achieving the electricity usage control of each household, considering the amount of lifestyle information acquired by the history of electricity usage data from the perspective of privacy preserving and data monopolism of companies.

Privacy preservation of households must be well considered when the data are shared to analyze electricity usage through DTI4SFS. Data analysts may recognize what types of electrical appliances were operated at a specific time and by which household. Figure 5–1 and Figure 5–2 show example results of the analyses [54] [55]. These analyzing results represent that electricity usage data contain lifestyle of the households. This technique is known as Non–Intrusive (Appliance) Load Monitoring [56] [57]. This information would be useful for criminals; for instance, a thief could estimate the family structure and reduce the risk of meeting the people who are living in the target’s household [58].

In this section, an anonymization method to share electricity usage while preserving data privacy is proposed. The method consists of a Self–Organizing Map (SOM) [59] for the anonymization. The proposed method was evaluated using two types of data captured from smart meters in different smart grids. The proposed method is adaptable in sharing any type of data while adjusting a special variable to maintain data accuracy according to the relevant analyses. Additionally, the proposed method can reduce information loss compared to another conventional method that anonymizes the data of each power supplier without data sharing.

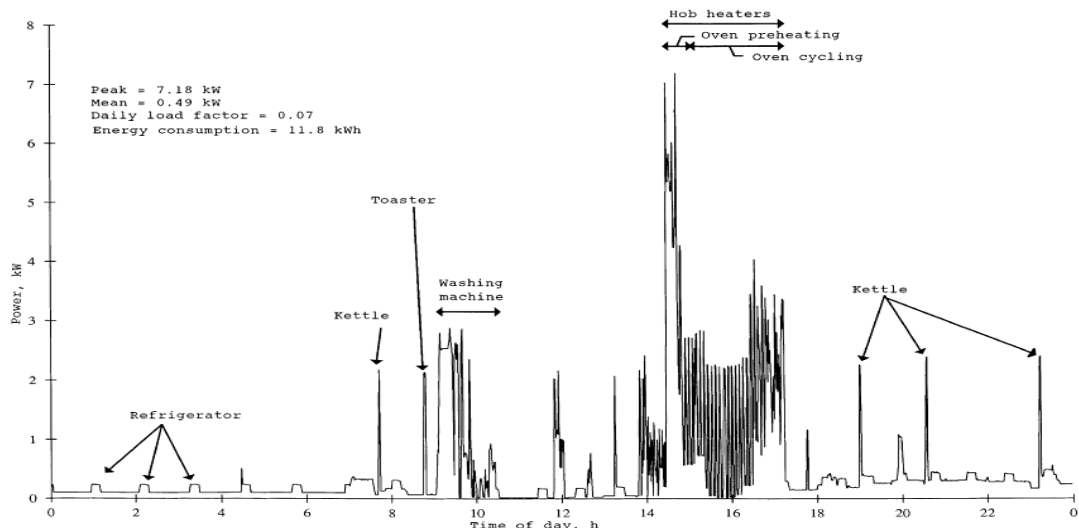


Figure 5–1 Electricity usage disclosed from electricity usage data (1) [54]

5 Anonymization method to share electricity usage data

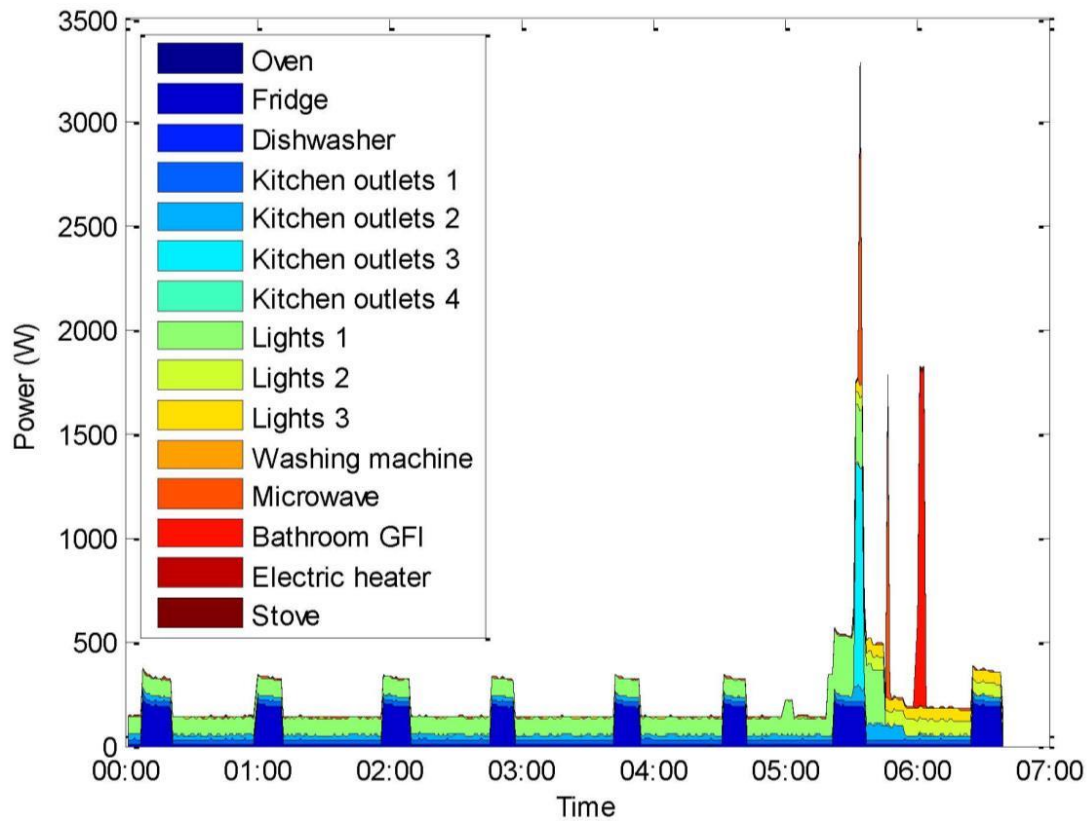


Figure 5-2 Electricity usage disclosed from electricity usage data (2) [55]

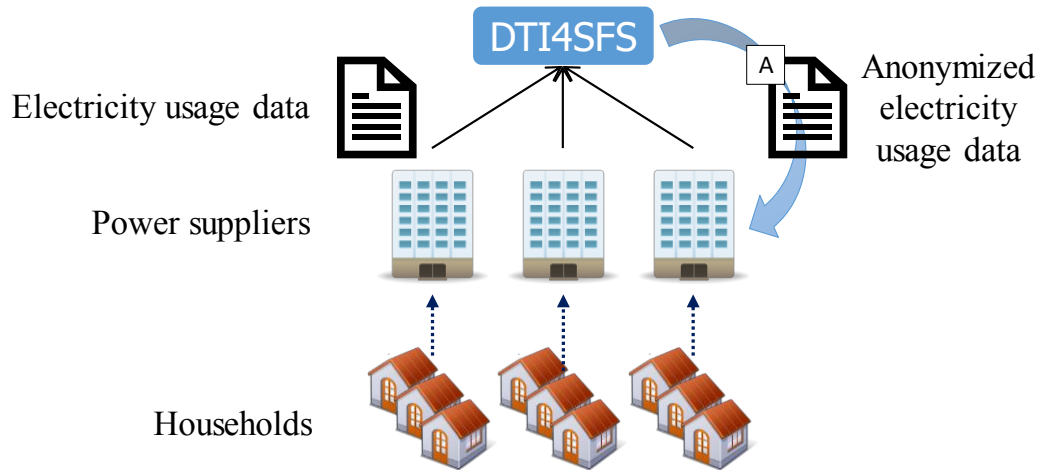


Figure 5-3 Overview of supposed situation in Section 5

In the supposed situation in this section, power suppliers are both data providers and data users of DTI4SFS while privacy information is related to households of the power suppliers. Publishing rules for the data sharing would be written by each power supplier while considering opinions from the households. Figure 5-3 shows the overview of the situation.

5 Anonymization method to share electricity usage data

5.1 Conventional approaches of PPDM for smart cities

There are some PPDM approaches to utilize electricity usage data while preserving data privacy as follows.

5.1.1 Approach based on homomorphic encryption

Homomorphic encryption is an approach of sharing data as mentioned in Section 2.1.1. This method provides encrypted calculation results without revealing the concerning values and equations. However, this approach requires high calculation costs. Moreover, the cost to manage encryption keys would be high when many smart meters are used. Guan et al. proposed a method to aggregate electricity usage data from smart meters in a smart grid using homomorphic encryption [60]. The method proposes a control center that is allowed to obtain the aggregated electricity usage data. The control center must pay a higher cost to use homomorphic encryption and to manage the encryption keys. Even if the costs are acceptable, several problems still exist for the DR services. The first problem is the data aggregation process. The control center in Guan's system cannot obtain any type of individual smart meter data, such as a pattern or cluster, by classification. This feature makes it difficult to issue the appropriate control commands to each household. Furthermore, the method does not allow the sharing of electricity usage data among power suppliers and others.

5.1.2 Perturbation based approach

Another approach is by randomizing values, i.e. perturbation as mentioned in Section 2.1.2. This approach replaces values in the data with alternative values to prevent disclosing the original value. The alternative values are generated by adding noise to the original values, or by randomly selecting other records in the data. A data user uses the data while estimating the original value using statistic methods [61]. Although this approach has an advantage in its wide application for various types of data, it lacks in preserving privacy because the original data can be estimated. Kursawe et al. propose some protocols to aggregate data in smart grids while preserving data privacy [62]. In this protocol, data providers add noise to data before the aggregation, and the aggregator obtains the sum of electricity usage by subtracting the noise. However, the aggregator only obtains the sum of the data [60].

5.1.3 Anonymization based approach

Anonymization generalizes the unique record to prevent the identification of the corresponding data and to preserve the privacy as described in Section 2.1.4, 2.2, and 2.3. k -member clustering is one of the anonymization algorithms of data clustering [63]. It creates clusters of similar records by generalizing records. This generalization maintains the number of records of each cluster greater than or equal to k . Subsequently, values in the same cluster are replaced with common values to satisfy the k -anonymity. If the anonymization result holds the information about the electricity usage, k -member clustering would be suitable to share electricity usage data.

5.2 Requirements for the anonymization

There is a problem to use k -member clustering to anonymize electricity usage data. The problem is that how to convert electricity usage data to the appropriate form for the k -member clustering while keeping data quality required by data users. Namely, a data clustering method to classify time-series data according to the data patterns is required.

Additionally, it is better if the data clustering method does not require a high-performance calculation environment. Since electricity usage data have locality, the anonymization does not need to be processed at the center of the proposed infrastructure. In other words, the anonymization process of the electricity usage data can be done in each area. Therefore, it is better if the clustering method can work on normal computers such as desktop machines or network appliances that any accelerating system for the calculation is not attached. This study adopts SOM for the clustering of the time-series data to fulfill their requirements.

5.3 Self-organizing map (SOM)

SOM is one of the unsupervised machine learning techniques based on artificial neural network and known as a clustering algorithm of time-series data [64]. Unsupervised machine learning techniques do not require training data whereas supervised machine learning techniques require the training data. The training data are referred by supervised machine learning techniques in the training step to learn how to correctly classify data. In contrast, unsupervised machine learning techniques enhance their accuracy of the classification or clustering by other methods such as iteration of their clustering step. For electricity usage data, unsupervised machine learning is appropriate due to difficulty of preparing the training data. In the other words, it is difficult to prepare the training data that clearly show the answer of the classification because there are many data patterns of electricity usage and the households change the patterns frequently. Each power supplier is only required to prepare the relevant electricity usage data.

SOM maps multidimensional records into two-dimensional nodes, which is called a map [59] (Figure 5-4). This feature reduces calculation cost after SOM, which means k -member clustering, because k -member clustering contains the repeated calculation step of the distances between the records. When applying SOM to electricity usage data, each component in the records holds electricity usage at the related timestamp. Since SOM creates clusters of given electricity usage data based on their data patterns, SOM can create the clusters while keeping information of the change of the electricity usage. In addition, the SOM allows the use of any number of record dimensions, although all records must be of the same number of dimensions, i.e., the SOM is executable for any electricity usage data if all data are acquired in the same period. Furthermore, the SOM does not require the number of clusters or the number of records in each cluster. It can ignore the size of the record of each power supplier for data sharing, and this is its unique and superior feature [65].

SOM can be divided into two phases as follows. First, a SOM randomly fills the node weight

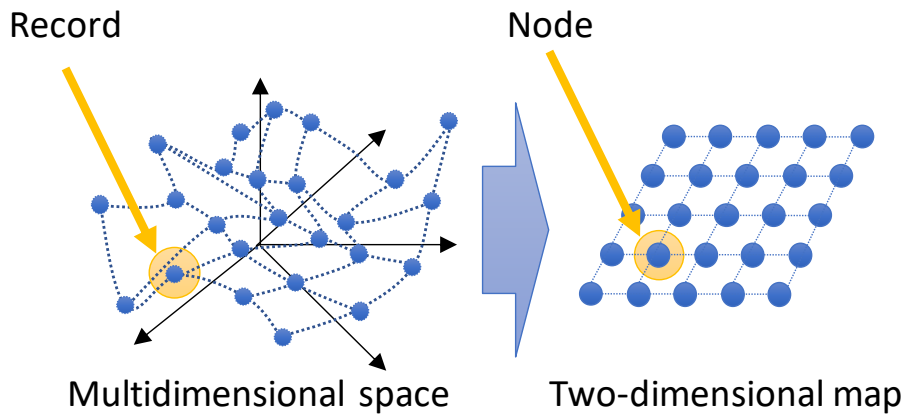


Figure 5-4 Self-organizing map

vectors of nodes in a map as initial values. The learning phase repeatedly proceeds the two-step executions below. The first step looks for the best matching unit (BMU) of each record. The BMU is the most similar node to a record. The second step is to update the weight vectors of the nodes in the neighborhood of the BMU and the BMU itself by pulling them closer to the input vector. These steps are repeated until the updated differences become less than a threshold λ .

In this section, the SOM-based data sharing method is proposed to preserve privacy. One of the laboratory members previously proposed a method to collect electricity usage data that uses the SOM to share data while considering data privacy [66], which is a novel technique using the SOM for PPDM. The method proposed in this section is advanced version of this previous study while focusing on utilization for DR. For these reasons described in Section 5.3, the SOM is used to extract the features of electricity usage in the proposed method.

In the previous method, the learning process of the SOM is executed repeatedly among power suppliers. The original method has a security problem. Power suppliers can tamper with the given SOM map. Moreover, the original method has a problem regarding calculation cost because it is difficult to parallelize the learning process. In addition, the proposed method improves the accuracy because it gives a higher weight to the peak power consumption to make the learning process effective for the DR services. In order to take the weight, peak time must be forecasted. The proposed method forecasts the time by using a model of time series analysis called seasonal autoregressive integrated moving average (SARIMA) model.

5.4 Time series analyses

Time series analyses analyze time series data while supposing that values in the data are related to their timestamps [67]. Time series data can be categorized into two types that are stationary time series data and non-stationary time series data. Time series data which statistical property are not changed are categorized to the stationary time series data.

Stationary time series data can be represented by autoregressive model (AR) when each of the

5 Anonymization method to share electricity usage data

values are affected by noise in the past. Equation 5-1 shows the AR where y_t and ε_t are the value and the white noise at t , respectively. The a is called autoregressive coefficient.

$$y_t = a_0 + \sum_{i=1}^p a_i y_{t-i} + \varepsilon_t \quad 5-1$$

Equation 5-1 can change the form as shown by equation 5-2. Since this form represents that time-series data depend on noises related to the past values, it is called moving average model (MA). The q -th order MA model can be represented by equation 5-3. The b is called moving average coefficient.

$$\begin{aligned} y_t &= \varepsilon_t + a_1 y_{t-1} \\ &= \varepsilon_t + a_1 \varepsilon_{t-1} + a_1^2 y_{t-2} \\ &= \varepsilon_t + a_1 \varepsilon_{t-1} + a_1^2 \varepsilon_{t-2} + a_1^3 y_{t-3} \\ &= \varepsilon_t + a_1 \varepsilon_{t-1} + a_1^2 \varepsilon_{t-2} + a_1^3 \varepsilon_{t-3} + \dots \end{aligned} \quad 5-2$$

$$\begin{aligned} y_t &= \varepsilon_t - b_1 \varepsilon_{t-1} - b_2 \varepsilon_{t-2} - \dots - b_q \varepsilon_{t-q} \\ &= \varepsilon_t - \sum_{i=1}^q b_i \varepsilon_{t-i} \end{aligned} \quad 5-3$$

To represent property of time series data from autoregression and moving average aspects, another model with combination of AR and MA has been proposed. This model is called ARMA. ARMA that consists of p -th order AR and q -th order MA can be represented by equation 5-4.

$$y_t = \sum_{i=1}^p a_i y_{t-i} - \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t \quad 5-4$$

Although ARMA is the model for stationary time series data, ARMA can be applied to non-stationary time series data when focusing on differences between neighbor values. Equation 5-5, 5-6, and 5-7 represent the first, second, and d -th order of the difference using lag operator L of $y_{n-1} = Ly_n$, respectively. The ARMA considering the differences represented by equation 5-8 is called autoregressive integrated moving average (ARIMA).

$$\Delta y_t = y_n - y_{n-1} = y_n - Ly_n = (1 - L)y_n \quad 5-5$$

$$\begin{aligned} \Delta^2 y_t &= \Delta y_t - \Delta y_{t-1} = (1 - L)y_n - (1 - L)y_{n-1} \\ &= (1 - L)y_n - (1 - L)Ly_n \\ &= (1 - L)^2 y_n \end{aligned} \quad 5-6$$

$$\Delta^d y_t = (1 - L)^d y_n \quad 5-7$$

5 Anonymization method to share electricity usage data

$$\Delta^d y_t = \sum_{i=1}^p a_i \Delta^d y_{t-i} - \sum_{j=1}^q b_j \Delta^d \varepsilon_{t-j} + \Delta^d \varepsilon_t$$

5-8

ARIMA can be applied to another type of non-stationary time series data when the time series data has cyclic variation. For instance, when the cycle of the variation i.e. season is one year, the average of the differences between the values related to the same timestamp of the different years is nearly zero while the differences are stationary time series data. Such model is called SARIMA model. The seasonal difference $\tilde{y}_t = y_t - y_{t-s}$ can be represented by equation 5-9 where P is the integral multiple of the season s . u_t in equation 5-9 is the error term. Φ and Θ are the parameters.

$$\tilde{y}_t = \sum_{i=1}^P \Phi_i \tilde{y}_{(t-s-i)} - \sum_{j=1}^Q \Theta_j u_{(t-s-j)} + u_t$$

5-9

The relationship between u_t and ε_t can be represented by equation 5-10 using parameters φ and θ . These two equations represent the SARIMA model.

$$u_t - \varphi_1 u_{t-1} - \dots - \varphi_p u_{t-p} = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_{t-q}$$

5-10

5.5 Proposed method to share electricity usage data

5.5.1 Procedure to acquire shareable data

The proposed method preserves privacy while converting records of electricity usage to a map of two-dimensional nodes created by the weighted SOM (WSOM) that is proposed in this study. After the WSOM repeatedly learns to obtain a node map of the electricity usages, it compares the input data with the temporal result of the learning to obtain the next temporal result. The input data are sets of records of electricity usage captured by a power supplier. It is enough to exchange the WSOM to obtain the characteristics of electricity usage without exchanging raw datasets of usage records obtained by power suppliers.

The proposed method consists of three steps: mapping step, gathering step and counting step. Each power supplier first obtains a node map by using the WSOM in the mapping step. The obtained node maps are put together in a node map of all the power suppliers in the gathering step. Since the mapping process and the gathering process are divided into two steps, the mapping process can be done by each of the power suppliers. In other words, the portion of the procedure can be parallelized to distribute the load of the calculation. The node map does not have information on the number of households belonging to each node in the node map. This information is added during the counting step. The node map is shown as a cluster of groups, and each group consists of multiple electricity usage data of the households. An attacker cannot identify electricity usage of her/his target household because it is already statistical data. Figure 5-5 illustrates the flow of the proposed method.

5 Anonymization method to share electricity usage data

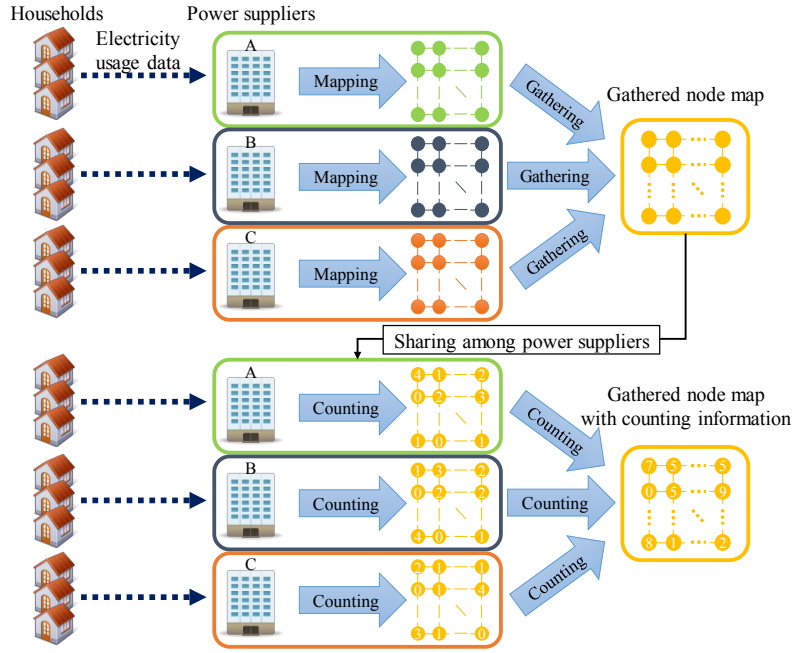


Figure 5-5 Flow of the proposed method to share electricity usage data

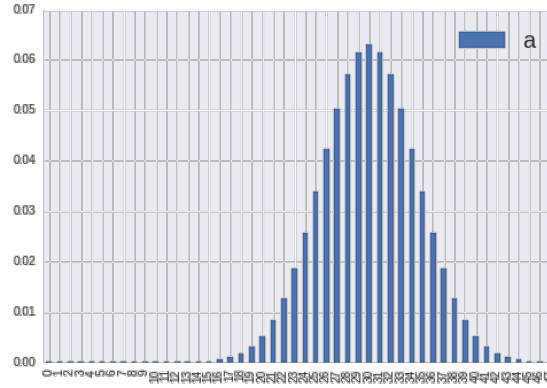


Figure 5-6 Example of the normal distribution function $a(t)$ ($p = 30$, $T = 48$)

5.5.1.1 Mapping step

The mapping step can be divided into three sub-steps. The first sub-step is executed only once, and the other sub-steps are executed iteratively. The first sub-step is the initialization of the map. This sub-step randomly resolves the initial positions and initial weight vectors of the nodes in the map. The second sub-step is to find the most similar BMU. The BMU for each record is calculated using equations 5-11 and 5-12, where $w_i(j)$ represents an i -th node and learns j times using the WSOM, t represents the sequential timestamp of the relevant record x where $1 \leq t \leq T$, and $a(t)$ is a normal distribution function that enhances the accuracy of learning around timestamp p . Figure 5-6 shows the example of $a(t)$ when p and T are 30 and 48, respectively. The accuracy around p compared with records of other timestamps increases when

5 Anonymization method to share electricity usage data

a small value is set to the deviation σ , and vice versa. When considering the DR, p should be at the peak of electricity usage. Therefore, the WSOM allows analysts of electricity usage to adjust the importance of the accuracy around p by σ according to their purposes. To set a small value to σ is proper for analyses that prioritize electricity usage around p , whereas a large value for σ is proper for analyses that prioritize all electricity usage times. Trials in using the WSOM beforehand are necessary to set the appropriate value of σ . The third sub-step is to update the nodes around the BMU. Equation 5-13 represents how the nodes are to be updated, where J denotes the maximum number of j 's, and $d_{BMU,i}$ denotes the Euclidean distance between the BMU and the i -th node.

$$BMU = \underset{i}{\operatorname{argmin}} \left(\sqrt{\sum_{t=1}^T a(t)(x(t) - w_i(t))^2} \right) \quad 5-11$$

$$a \sim N(p, \sigma^2) : a(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-p)^2}{2\sigma^2}} \quad 5-12$$

$$w_i(j+1) = w_i(j) + \left(1 - \frac{j}{J}\right) e^{-\frac{d_{BMU,i}^2}{2\sigma(j)^2}} \quad 5-13$$

5.5.1.2 Gathering step

In the gathering step, a node map of all the power suppliers is assembled from the node maps obtained in the mapping step. The mapping step method is again used for the assembling. This node map is shared among power suppliers.

5.5.1.3 Counting step

In the counting step, power suppliers check how many households are related to each node in the node map generated in the gathering step. The easiest way of sharing the counting information is by using a database. However, this is not the best way when considering the privacy risks of sharing counting information because each power supplier could identify the information given by other power suppliers. Three options for the counting are assumed as follows.

Using homomorphic encryption is the first option. In this option, the counting result is kept secret. Although the calculation cost of homomorphic encryption tends to be high, in this case, the counting step only requires adding values, which makes the calculation cost of homomorphic encryption much lower than usual. Therefore, homomorphic encryption is suitable compared to other methods that are appropriate for calculation cost saving, such as Paillier encryption [68] and the additive El Gamal encryption [69].

The second option is by making the power suppliers add the information in the rotation. The first power supplier adds an initial value to a counter for a node in the node map respectively

5 Anonymization method to share electricity usage data

while the power supplier only knows the initial values. For the next step, all power suppliers add the number of households to relevant counters in the rotation. Finally, the first power supplier subtracts the initial values from the respective counters to finalize the result. Although this option also preserves the data privacy of households, it is not safer than the first option when power suppliers cooperate to reveal the information added from a power supplier. The final power supplier is also able to know the initial values. For instance, the values added by the first power supplier could be revealed if other power suppliers subtract the values added by them from the relevant counter.

The third option is to ask a trustworthy third-party agency, such as the aggregator, to manage the counting instead of a database. A power supplier first obtains the node map assembled in the gathering step and counts the number of households for their relevant node. The counting result is sent to the agency, who publishes the counting result—which is the sum of all the information obtained from all power suppliers—but does not publish each information.

5.5.2 Forecast of peak electricity usage

To define p in equation 5-12, the proposed method requires forecasts of the peak time of electricity usage. The proposed method uses a SARIMA model for the forecasting because this model can be constructed even from values captured from seasonal time-series data such as electricity usage data captured from smart meters [67]. In the proposed method, the electricity usage in one day is forecasted by the SARIMA model while the electricity usage until seven days before the day is used as the input values. In this study, the parameters of the SARIMA model were defined by using Augmented Dickey-Fuller (ADF) test and Akaike's information criterion (AIC) [70].

5.6 Evaluation

Two datasets of the smart meter are used to evaluate the proposed method. One of the datasets was captured in a smart city located in Kawasaki city, Japan (D_K). The other dataset, named the Irish smart meter dataset, was provided by the commission for regulation (D_I) [71]. The numbers of households in each dataset are 53 and 1,000, respectively. The timestamp period of the data is 30 min in both datasets. The evaluation environment including the proposed method was implemented using Python 2.7.

5.6.1 Information loss

In this evaluation, k -member clustering is assumed to be used to anonymize the electricity usage data to publish the data while preserving data privacy. As mentioned in Section 2.4, anonymization causes information loss, which depends on data property. A comparison of two different anonymization methods is carried out to evaluate the information loss when using the same datasets in order to prevent the influences of data property [72] [73]. The two situations of k -member clustering were compared to evaluate the information loss in the proposed method.

5 Anonymization method to share electricity usage data

Table 5-1 Rate of information loss (R_{IL}) in D_K
when n_p and k are set from 2 to 4 and 2 to 10, respectively (MapSize is 5×5)

n_p	k								
	2	3	4	5	6	7	8	9	10
2	1.55	1.23	1.15	1.09	1.05	1.06	0.99	1.04	1.04
3	1.45	1.13	1.08	1.03	1.03	0.96	0.99	1.00	1.01
4	1.38	1.15	0.98	1.02	0.99	0.99	0.90	0.94	0.96

Table 5-2 Rate of information loss (R_{IL}) in D_K
when n_p and k are set from 2 to 4 and 2 to 10, respectively (MapSize is 10×10)

n_p	k								
	2	3	4	5	6	7	8	9	10
2	1.20	1.05	1.05	1.00	1.00	1.03	0.96	1.04	1.04
3	1.13	0.98	0.97	0.98	0.98	0.95	0.95	1.00	1.05
4	1.06	0.98	0.90	0.96	0.95	0.97	0.88	0.92	0.96

Table 5-3 Rate of information loss (R_{IL}) in D_I
when n_p and k are set from 5 to 15 and 2 to 20, respectively (MapSize is 10×10)

n_p	k									
	2	4	6	8	10	12	14	16	18	20
5	1.62	1.14	1.03	0.97	0.94	0.92	0.92	0.90	0.91	0.89
10	1.51	1.07	0.97	0.93	0.90	0.89	0.89	0.88	0.86	0.83
15	1.48	1.04	0.96	0.92	0.88	0.87	0.84	0.86	0.81	0.82

Table 5-4 Rate of information loss (R_{IL}) in D_I
when n_p and k are set from 5 to 15 and 2 to 20, respectively (MapSize is 20×20)

n_p	k									
	2	4	6	8	10	12	14	16	18	20
5	1.45	1.03	0.94	0.90	0.88	0.88	0.88	0.87	0.88	0.86
10	1.32	0.95	0.88	0.85	0.83	0.84	0.85	0.84	0.83	0.81
15	1.28	0.92	0.86	0.83	0.81	0.81	0.80	0.82	0.78	0.80

For both situations, the electricity usage data of all power suppliers are anonymized by k -member clustering. The difference in the situations is the input data of the clustering. In one situation, the input of the k -member clustering is the electricity usage data including all power suppliers generated by the proposed method. In the other situation, the data of each power supplier are anonymized, respectively. The information loss of the situations (IL_P and IL_I , respectively) is calculated using the equation 5-14, where N denotes the number of records, and c_j and x_j represent the value after and before anonymization, respectively. Equation 5-15 represents the rate of the information loss in the two situations.

$$IL = \frac{1}{N} \sum_{j=1}^N |c_j - x_j|$$

5-14

5 Anonymization method to share electricity usage data

$$R_{IL} = \frac{IL_P}{IL_I}$$

5-15

Table 5-1 and Table 5-2 represent R_{IL} in D_K while k is defined from 2 to 10, and the number of power suppliers n_p is defined from 2 to 4. The map sizes of the WSOM are 5×5 and 10×10 , respectively. Table 5-3 and Table 5-4 represent R_{IL} in D_I while k is defined from 2 to 20, and the number of power suppliers n_p is defined from 5 to 15. The map sizes of the WSOM are 10×10 and 20×20 . Although k would not set to small number such as 2 in real situation due to the paucity of anonymity, k was defined from 2 just for the evaluation.

According to the four tables, IL_P is smaller than IL_I when the values of n_p and k are large in both datasets. Compared with the conditions, IL_P is larger than IL_I when k is defined to a small value, such as 2. The evaluation result shows that the information loss of the proposed method is reduced when n_p and k are large. In particular, the information loss is reduced up to 22% in Table 5-4 when k and n_p are set to 18 and 15, respectively.

5.6.2 Accuracy of value while considering DR

As mentioned in equation 5-12, the accuracy of the proposed method depends on $a(t)$, and its strength can be adjusted by σ . Figure 5-7 and Figure 5-8 represent the amount of mean absolute error (MAE) at all times (MAE_{all}) and at the peak time (MAE_{peak}) while shifting σ^2 from 0.1 to 100.0. MAE_{all} and MAE_{peak} are calculated by equations 5-16 and 5-17, respectively. The amounts of error in Figure 5-7 and Figure 5-8 are measured while D_K and D_I are used, respectively. In addition, the amounts of error when no weight was given ($a(t) = 1$) are also measured, and represented in these two figures.

$$MAE_{all} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T |w_n(t) - x_n(t)|$$

5-16

$$MAE_{peak} = \frac{1}{NT} \sum_{n=1}^N |w_n(p) - x_n(p)|$$

5-17

According to Figure 5-7 and Figure 5-8, the accuracy during peak times is high when σ^2 is small whereas the accuracy at all times is lower than during peak times. There is a trade-off between the accuracy during peak times and at all times. This trade-off is shown from the result when σ^2 is set to 1.0 in D_K . According to Figure 5-7, the result has a 40% lower MAE_{peak} but a 56% higher MAE_{all} than the no-weight result. In addition, the balance in the trade-off depends on the relevant dataset. The dependency is also shown from the results in Figure 5-7 and Figure 5-8, where σ^2 is set to 1.0. In contrast to D_K , the result of D_I has a 60% lower MAE_{peak} but a 40% higher MAE_{all} than the no-weight result. Therefore, σ^2 must be defined based on the required accuracy for each dataset. When the supposing DR service for D_K allows an MAE_{all} of up to 0.08 kWh, choosing 1.0 as σ^2 will minimize the MAE_{peak} .

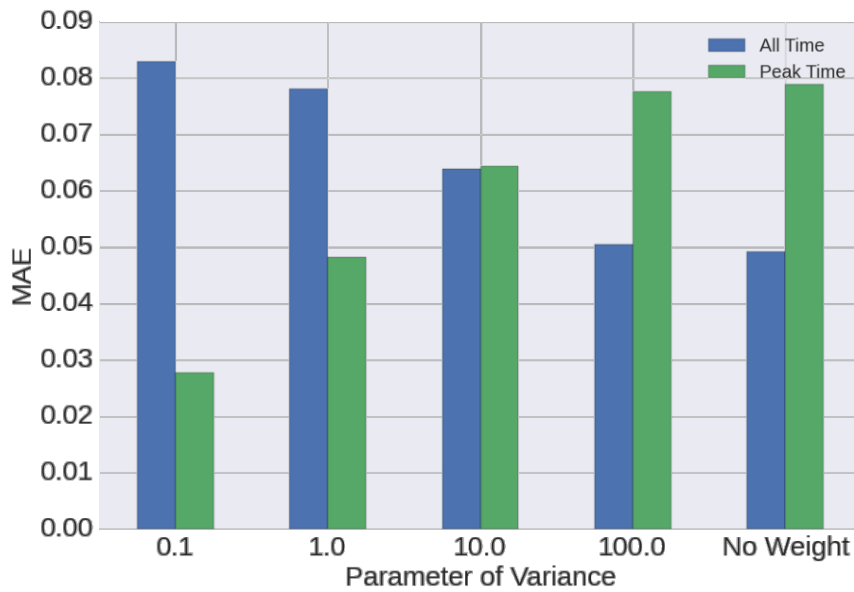


Figure 5-7 Amount of mean absolute error (*MAE*) in all and peak time in D_K while shifting parameter of variance (σ^2) is from 0.1 to 100.0

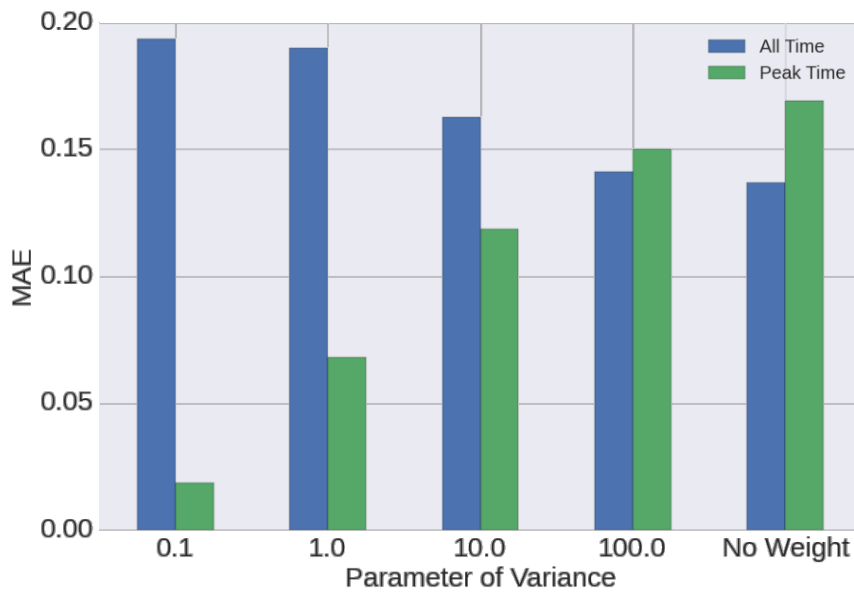


Figure 5-8 Amount of mean absolute error (*MAE*) in all and peak time in D_I while shifting parameter of variance (σ^2) is from 0.1 to 100.0

5.7 Summary

In this section, an anonymization method to share electricity usage while preserving data

5 Anonymization method to share electricity usage data

privacy is proposed. The proposed method uses the self-organizing map and allows power suppliers to share a node map of the electricity usage while allowing the raw data to be captured from the smart meters. The proposed method enables DTI4SFS to anonymize time-series data.

Our evaluation results show that the proposed method achieves the same level of k -anonymity with small information loss compared with other conventional method that anonymize data of each power supplier without data sharing. In particular, information loss is reduced by 22% when k and n_p are set to 18 and 15, respectively.

The evaluation result also shows that the electricity usage data shared by the proposed method can be used for the DR. The accuracy of the shared data can be adjusted, and a trade-off exists between the accuracy at all times and during peak times.

6 Watermarking method for anonymized data

In this study, DTI4SFS that is an anonymization infrastructure for PPDP using anonymization is proposed. The infrastructure allows data users to acquire anonymized data by having them request requirements such as data entries and privacy protection levels. Request constraints are defined by the data provider and are published with the data property by the infrastructure in advance. The infrastructure has a special anonymization procedure named one-directional anonymization that refers to previously published data. Such procedure prevents attacks that spoil the privacy protection by aggregating previously published data.

However, the proposed infrastructure still has a problem when it comes to the illegal republishing of data. The infrastructure provides the same data to some data users when the users request the same requirements to the infrastructure, and some of the users may republish the anonymized data without the permission of the data providers. In this case, the illegal data user should be identified when the illegally published data is accessed. This identification function will act as a deterrent to illegal republication. Anonymized data should provide the information pertaining to the official user of the data to identify illegal data republication. Therefore, published data should be unique regarding its data users. Watermarking is well suited for this purpose. Watermarking adds a watermark into the data by modifying the values of the data to identify the users who published the data. Figure 6-1 shows the supposed usage of watermarking in this study. In the proposed infrastructure, information related to each data user is watermarked to the anonymized data before the data publishing. When the illegally published data is found, the watermarked information will be extracted to identify the illegal data user. In the watermarking research domain, multimedia data such as audio data, picture data, video data, and text data are popular targets for watermarking [74] [75]. Despite the availability of watermarking, the conventional watermarking methods are not appropriate for use with anonymized data because they change the anonymity level and degrades the usefulness of the data.

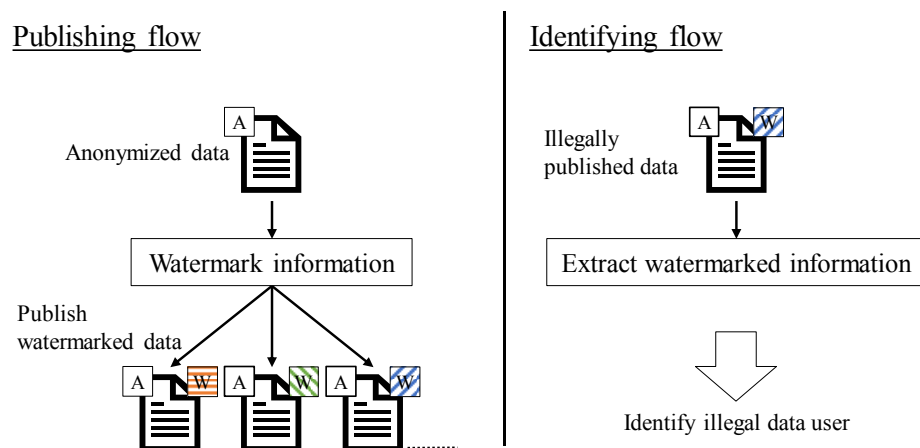


Figure 6-1 Supposed usage of watermarking in the proposed infrastructure

6 Watermarking method for anonymized data

Moreover, the type of anonymized data, which DTI4SFS deals with, is different from multimedia data. Multimedia data allows us to watermark information sequentially, because the data comprises ordered information, such as the elapsed time of the audio or video data. Additionally, multimedia data does not negatively influence the quality of the data when modified, given that multimedia data has data redundancy. Data such as audio data has high-frequency components that humans cannot hear. In contrast to multimedia data, records in anonymized data do not have a sequence, and their values do not have suitable positions for modification by watermarking.

Another issue is the revealing of watermarks, which is a common problem faced by all watermarking techniques. Revealing may occur when several anonymized datasets are generated and published from the same original data. Watermarking for anonymized data has to be tolerant of a modification that reveals the watermarks while simultaneously preventing the degradation of the anonymity level.

6.1 Related techniques of the proposed watermarking method

The proposed watermarking method utilizes several error correction techniques in conjunction with cryptography. This section describes the combined techniques.

6.1.1 Advanced encryption standard

The Advanced Encryption Standard (AES) is a symmetric-key encryption method and is a type of block cipher. Symmetric key encryption uses one secret key for both the encryption and decryption processes. The block cipher encrypts plaintext at intervals of several bits. Each interval is called a block; the general block length of AES is 128 bits. The length of ciphertext is multiple numbers of the block length.

AES outputs ciphertext that is generated from multiple blocks when the length of the plaintext is longer than the block length. The encryption process should not isolate the blocks to prevent attacks that replace parts of the ciphertext to modify its plaintext. Moreover, the key for encryption should be changed with every block encrypted; otherwise, attackers would be able to read the plaintext from a ciphertext without decryption. In other words, a block of ciphertext will show the same block of plaintext when the same key is used for the encryption. Several ways to connect blocks have been proposed to solve these problems such as Cipher Block Chaining (CBC) mode, Cipher-FeedBack (CFB) mode, Output-FeedBack (OFB) mode, and CounTeR (CTR) mode [76].

6.1.1.1 CBC mode

In CBC mode, neighbor blocks are connected by exclusive-OR operations. Figure 6-2 and Figure 6-3 illustrates the encryption and decryption flows, respectively. The encryption flow of the CBC mode takes an exclusive OR of a block of the plaintext and a block of the ciphertext encrypted from the previous block of the plaintext before every encryption process. The exclusive OR processes prevent the problem that could be happened when the same blocks of plaintext

6 Watermarking method for anonymized data

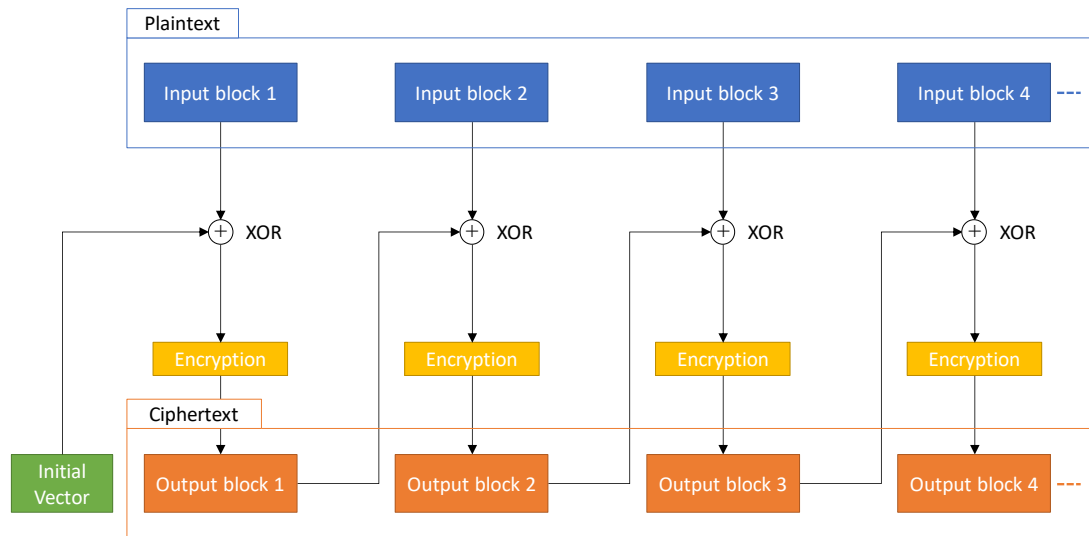


Figure 6-2 Encryption flow of CBC mode (refer to [76])

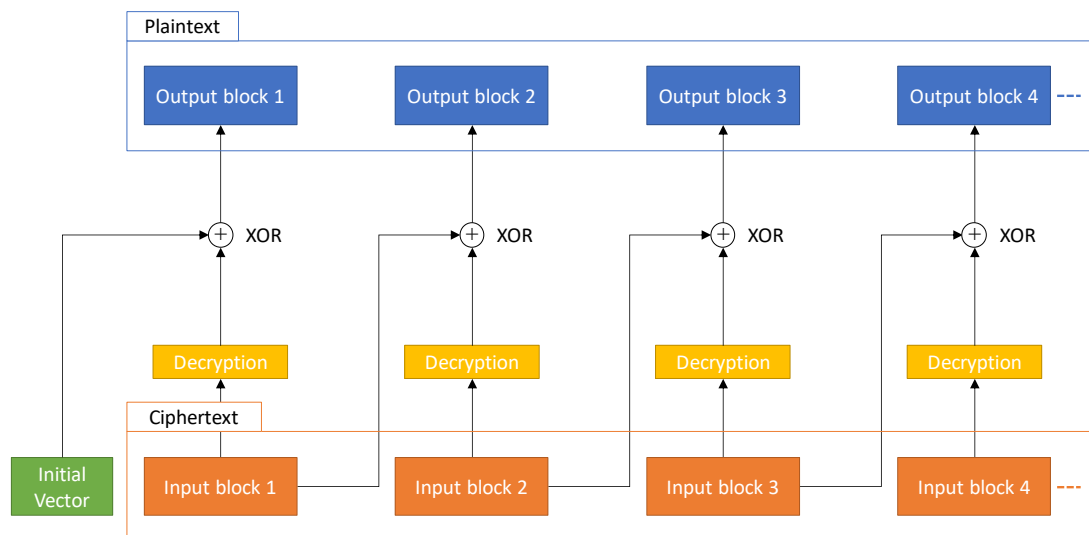


Figure 6-3 Decryption flow of CBC mode (refer to [76])

appear. CBC mode requires an initial vector (IV) taken to the exclusive OR of the first block of the plaintext. Ciphertext of AES with CBC mode is unique if the same combination of the key for the encryption and the IV is set to the same plaintext. IV is generally updated at every encryption by using a pseudorandom generator whereas the key for the encryption is not changed.

When a part of ciphertext is flipped from 0 to 1 and vice versa (bit flipping), the flipping affects two blocks which are the block containing the flipped bit and the neighbor block. When a bit string of a block in the ciphertext is changed, the change affects the changed block and blocks encrypted after the changed block. The affected blocks would fail to be decrypted.

6.1.1.2 CFB mode

Figure 6-4 and Figure 6-5 illustrate encryption and decryption flow of AES with CFB mode,

6 Watermarking method for anonymized data

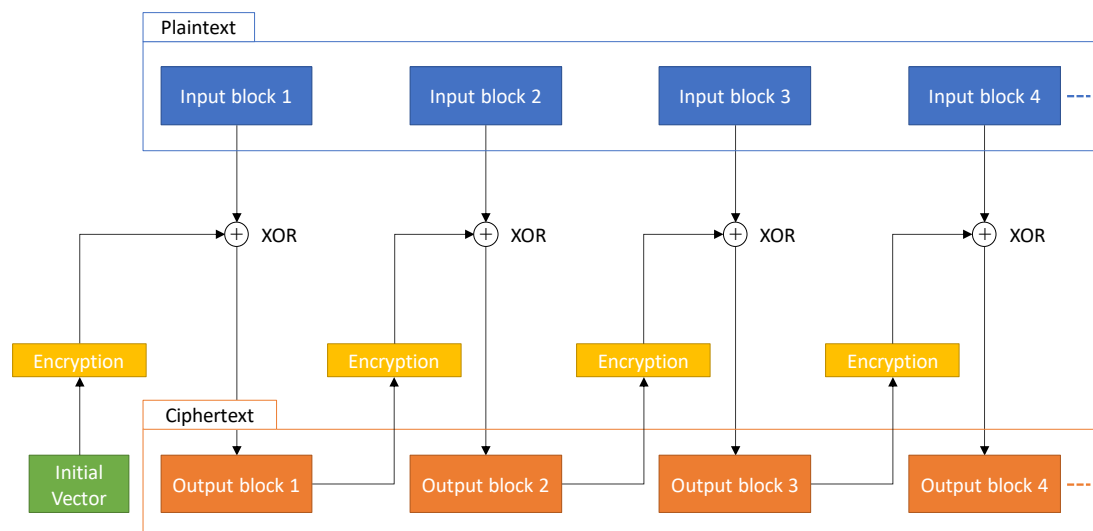


Figure 6-4 Encryption flow of CFB mode (refer to [76])

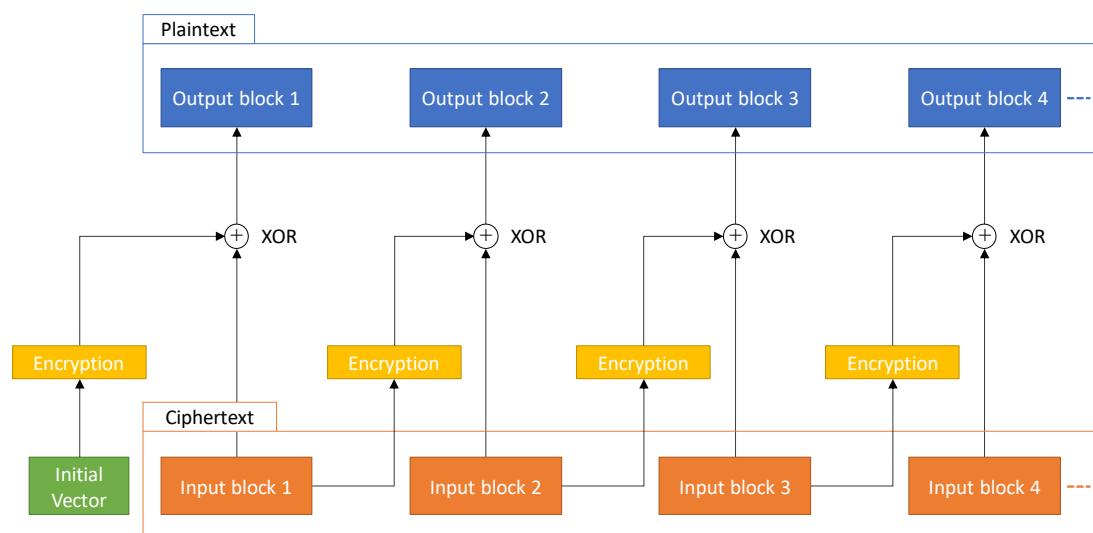


Figure 6-5 Decryption flow of CFB mode (refer to [76])

respectively. Compared with CBC mode, IV or a block of the previously output ciphertext is encrypted whereas the other parts of the encryption flow are same as CBC mode. On the other hand, the decryption flow of CFB mode uses the encryption process of AES instead of the decryption process.

Influence of the bit flipping and the change of block length are same as CBC mode. Bit flipping of a ciphertext encrypted by AES with CFB mode affects the block containing the flipped bit and its neighbor block. Change of bit string of a block in the ciphertext affects the changed block and blocks encrypted after the changed block.

6.1.1.3 OFB mode

Figure 6-6 and Figure 6-7 illustrate the encryption and decryption flows of AES with OFB

6 Watermarking method for anonymized data

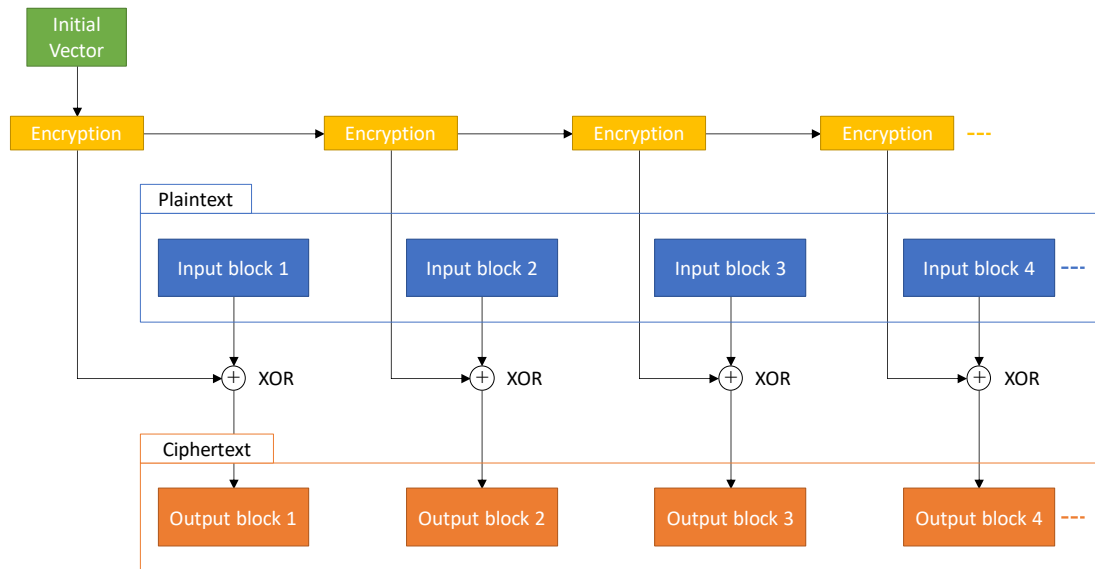


Figure 6-6 Encryption flow of OFB mode (refer to [76])

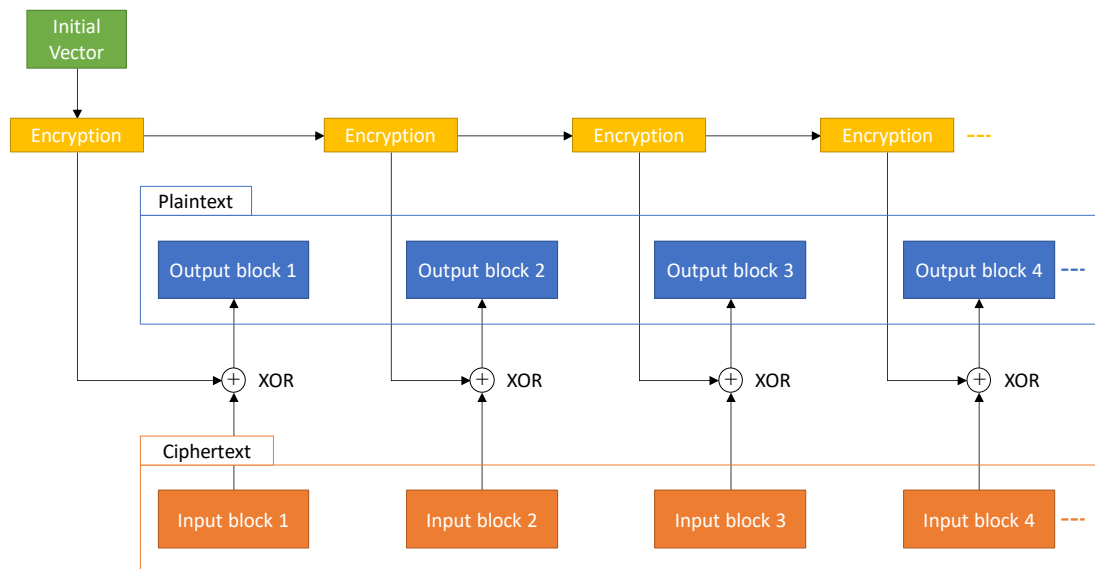


Figure 6-7 Decryption flow of OFB mode (refer to [76])

mode. In the encryption flow, IV is encrypted by AES iteratively before every exclusive-OR operation whereas the plaintext itself is not directly encrypted. The decryption flow consists of encryption processes of AES as same as CFB mode. Implementation cost of OFB mode is lower than the other mode because the decryption flow of OFB mode is same as the encryption flow. Moreover, the flows can be parallelized because the encryption process and the exclusive-OR operation are independent of each other. The parallelization is achieved when the all encryption processes of IV are done beforehand.

Influence of the bit flipping and the change of block length are less than CBC and CFB modes because each block in OFB mode is isolated whereas the encrypted IVs are not isolated. Namely,

6 Watermarking method for anonymized data

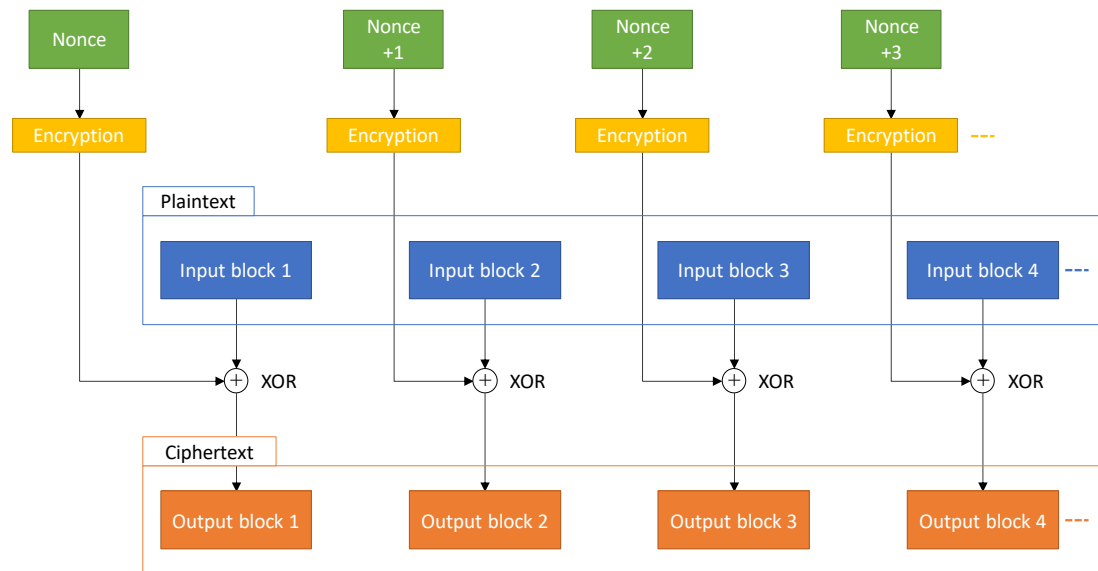


Figure 6–8 Encryption flow of CTR mode (refer to [76])

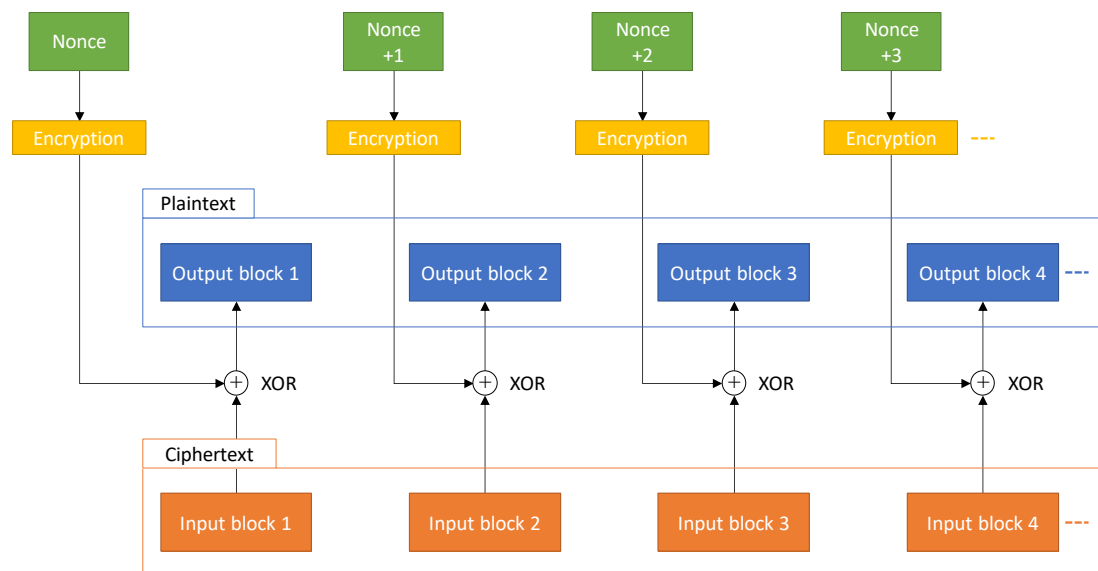


Figure 6–9 Decryption flow of CTR mode (refer to [76])

the bit flipping and the change of block length affect the flipped or changed block only. However, the position of the flipped bit on the ciphertext is same as its plaintext. Therefore, attackers can modify the plaintext without the decryption process.

6.1.1.4 CTR mode

AES with CTR mode requires a bit string named nonce instead of an IV. The role of nonces is similar to IVs. Figure 6–8 and Figure 6–9 illustrate the encryption and decryption flows of CTR mode, respectively. The nonce is counted up to prepare unique values to each block of the plaintext. Each of the prepared values is encrypted, and the block of the ciphertext is generated by an exclusive-OR operation of the encrypted value and the block of the plaintext. In contrast

6 Watermarking method for anonymized data

to OFB mode, the encryption processes of AES can be parallelized as well as the exclusive-OR operations. Influence of the bit flipping and the change of block length are same as OFB mode.

CBC mode and CTR mode are recommended by Ferguson et al. [77]. As shown in Section 6.1.1.1 and 6.1.1.4, one of the differences between CBC mode and CTR mode is the influence on neighboring blocks when the bit flipping happens. In CBC mode, when a bit in a block gets flipped, it affects the block and its two neighboring blocks which prevents them to provide the correct decryption result. In contrast, bit flipping in CTR mode only affects a single block.

Cryptography techniques should not give attackers information that facilitates their attacks, such as information that comes from the imbalance of the bits in the ciphertexts. Therefore, bits in ciphertexts should be well balanced. According to Matsuoka et al., the number of bits that express '1' is 50% in the ciphertext of AES [78]. The study shows that bits in the ciphertext of AES are well balanced.

6.1.2 Turbo code

The error-correction code enables one to recover the correct bit string, even when the bit string contains flipped bits, by adding a string of parity bits into the original bit string. Turbo code is one of the error-correction codes whose transmission efficiency is regarded to be among the most efficient in the data transmission domain [79]. In other words, turbo code requires shorter bit lengths to encode a bit string to achieve the same error-correction ability as other error-correction code methods. Figure 6-10 illustrates the flow of the encoding process. In the encoding process, the turbo code first interleaves the input data string, generating two parity bit strings from the original data string and the interleaved string using an internal encoder. Next, the generated strings are decimated alternately to reduce the total length of the two strings by half (the puncturing process). Although the puncturing process is irreversible, it is effective in reducing the length of a bit string that results from the encoding. Finally, the punctured parity bit string is connected to the original bit string. Figure 6-11 shows the puncturing process.

Figure 6-12 illustrates the overview of the decoding process. In the decoding process, the turbo code attempts to obtain the decoded result by repeatedly running an internal decoder. The

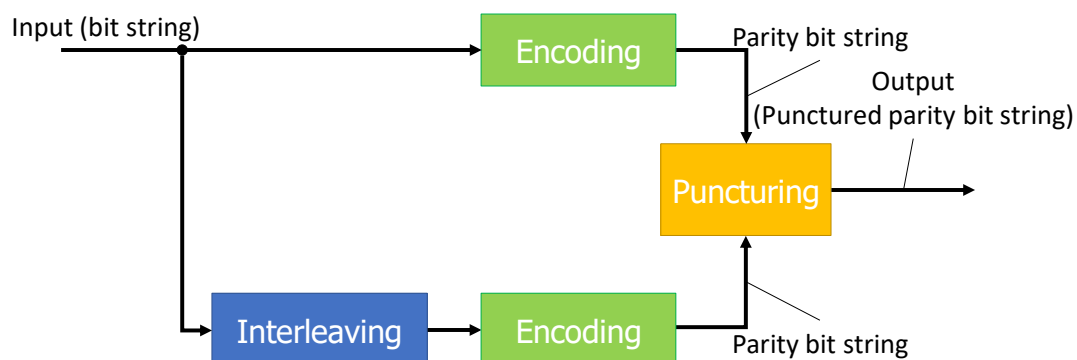


Figure 6-10 Flow of the encoding process of turbo code (refer to [79])

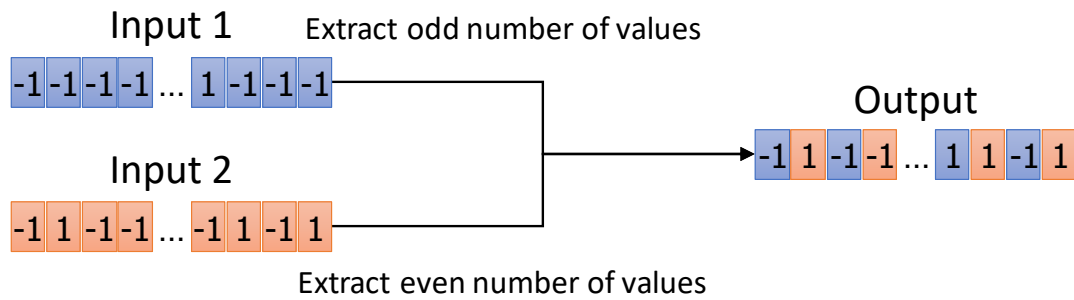


Figure 6-11 Puncturing process

Soft Output Viterbi Algorithm (SOVA) [80] is one such process algorithm. In this algorithm, a data string and a parity bit string are used for each running process. Additionally, a decoded result output from the previous running process is also input into the next running process. Prior to the decoding process, an encoded bit string is divided into a data string and a parity bit string. The two parity bit strings that were punctured during the encoding process are recovered from the parity bit string of the encoded bit string by using the intermediate values instead of the bits omitted during the puncturing process as shown in Figure 6-13. For each running process of the internal decoder, non-interleaved and interleaved data strings are used alternatively. The previous decoded result is also interleaved or de-interleaved before the next running process using the data string. The recovered parity bit string is selected according to the condition whether the data string is interleaved or not. The final decoding result will be completed by the iteration of the internal decoding.

6.1.3 Gray code converter

Gray code converts data values to make all Hamming distances of neighboring values equal to 1. Generally, when the Hamming distance is small, a value that includes a flipped bit expressing a similar value to the original value. Therefore, using the gray code with error-correction code enhances the error-correction ability of the code.

Figure 6-14 shows examples of the conversion. The gray code converter outputs an exclusive OR of an input bit string, and a bit string which was a one-bit shifted input bit string to the lower bit. For instance, when an input bit string is '101', the gray code converter outputs '111' that is an exclusive OR of '101' and '010'. As shown in Figure 6-14, Hamming distances of neighboring output values are 1.

6.2 Proposed watermarking method

6.2.1 Features of the proposed watermarking method

In this study, the proposed watermarking method is used to identify the person who republished

6 Watermarking method for anonymized data

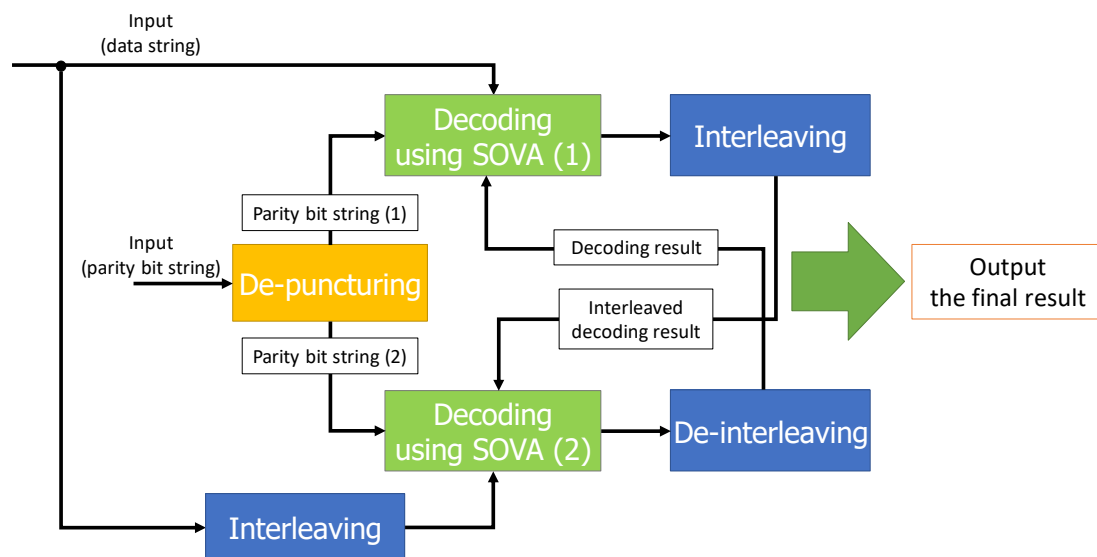


Figure 6–12 Flow of the decoding process of turbo code (refer to [79])

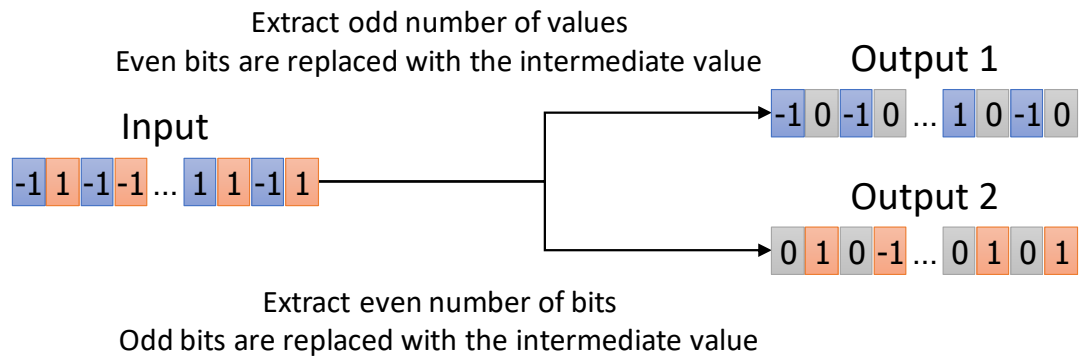


Figure 6–13 Process of recovering from punctured string

the anonymized data received from DTI4SFS without authorization. The watermarking method should allow only authorized people to access the watermarked information (access control). Also, watermarked information should be able to be extracted even if the anonymized data containing that watermarked information is modified (robustness). Moreover, the proposed watermarking method should include protection from attacks where multiple malicious data users attempt to extract watermarked information by comparing the anonymized data received by each of the data users (a collision attack).

In the proposed watermarking method, watermarked information is encrypted before being watermarked, and it uses AES for access control. Access control can be achieved because only the people who are authorized to access the watermarked information know the secret key of the encryption.

Use of AES is also effective to protect against collusion attacks. Attackers in collusion attacks attempt to extract watermarked information by comparing multiple instances of the published anonymized data. Even when the proposed watermarking method expresses the watermarked

6 Watermarking method for anonymized data

Input value		Shifted value of input	=	Output value
1 0 1	\oplus	0 1 0	=	1 1 1
1 1 0	\oplus	0 1 1	=	1 0 1
1 1 1	\oplus	0 1 1	=	1 0 0

Figure 6-14 Converting examples of gray code

information in the form of binary data, when the values of the anonymized data are modified, attackers cannot identify whether the differences express ‘0’ or ‘1’ from a statistical point of view if the probability of occurrence of ‘0’ and ‘1’ in the binary data is the same. As mentioned in Section 6.1.1, the probability is the same in AES ciphertexts. Therefore, the proposed watermarking method can protect watermarked information from collusion attacks.

The proposed watermarking method should also consider another type of attack whereby attackers try to delete or degrade the watermarked information by changing the values of published data (a distortion attack). Distortion attacks can be treated as noise added to the data. In this study, turbo code is used to protect watermarks from distortion attacks, since the length of the parity bit string that the code requires is shorter than other error-correction code techniques. Given that the degree of modification for watermarking is proportional to the length of the watermarked bit string, using turbo code is one of the best approaches to minimize degradation of the value of the anonymized data resulting from watermarking. The proposed watermarking method encodes information that will be watermarked into anonymized data after AES encryption. Additionally, the encoded information is converted using a gray code converter to enhance the error correction ability of turbo code.

6.2.2 Entire flow of the proposed watermarking method

Figure 6-15 illustrates the entire flow of the proposed watermarking method. The upper side and the bottom side of the figure show the watermarking and extracting flows, respectively. In the watermarking flow, a plaintext is first encrypted to ciphertext by AES with the block length of the encryption being 128 bits. Next, the ciphertext is encoded by the turbo code. The encoding adds a parity bit string whose length is same as the ciphertext. Finally, the encoded bit string is watermarked to the anonymized data by modifying the values of that data. The encoded bit string is converted into gray code just before the modification.

In the extracting flow, the proposed watermarking method first extracts a bit string from the anonymized data that has been republished without authorization. The extraction involves the inverse conversion of the gray code. Next, the extracted bit string is decoded by the turbo code, and finally, the decoded string is decrypted by AES. In the proposed watermarking method, CTR mode is selected for use by AES since the influence of distortion attacks is small.

A systematic convolutional code was implemented as the internal encoder of the turbo code. The code holds a current state and decides the next state according to an input bit and the

6 Watermarking method for anonymized data

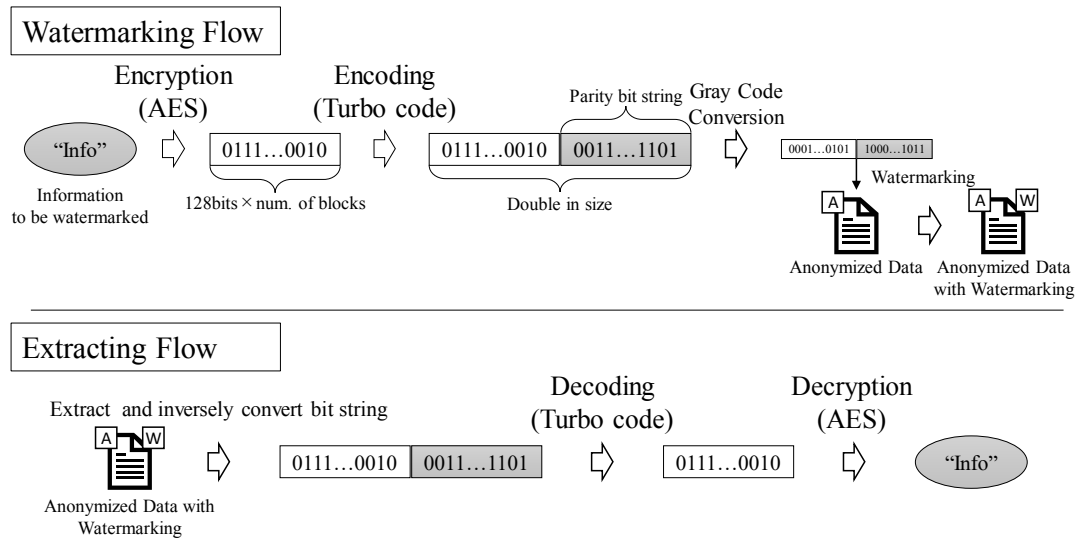


Figure 6-15 Entire flow of the proposed watermarking method

current state. A parity bit is an output when the state changes. Figure 6-16 shows a trellis diagram of the implemented code. The initial state is S1. When a bit string ‘011’ is input, ‘010’ is output as a parity bit string while its state changes from S1 to S2 via S3. The implemented turbo code used SOVA for the decoding process.

Figure 6-17 illustrates the interleaving process that was implemented for the turbo code. The interleaver divides an input bit string into parts at 3-bit intervals. The parts of the bit string are arranged in row directions. An interleaved bit string is created by joining the arranged parts along the column direction. The reverse process of the interleaving was also implemented as the de-interleaver.

6.2.3 Watermarking process of the proposed method

In the proposed watermarking method, the process that watermarks the information into anonymized data consists of two sub-processes: a grouping process and a modification process of anonymized values. The grouping process first sorts records of anonymized data and then makes record groups so that each group includes all records that have the same values of the sorted items. The grouping process was implemented to watermark information sequentially. Otherwise the sequential order for the watermarking would be broken by changing the order of the records since anonymized data includes records that have the same values for privacy protection.

In the modification process, values are modified to express the watermarked information. The number of modified records in each record group expresses the watermarked information. Each record group expresses a part of the information when the information is larger than the number of records in the record group because the bit length of the part is as large as the largest the record group can express.

6 Watermarking method for anonymized data

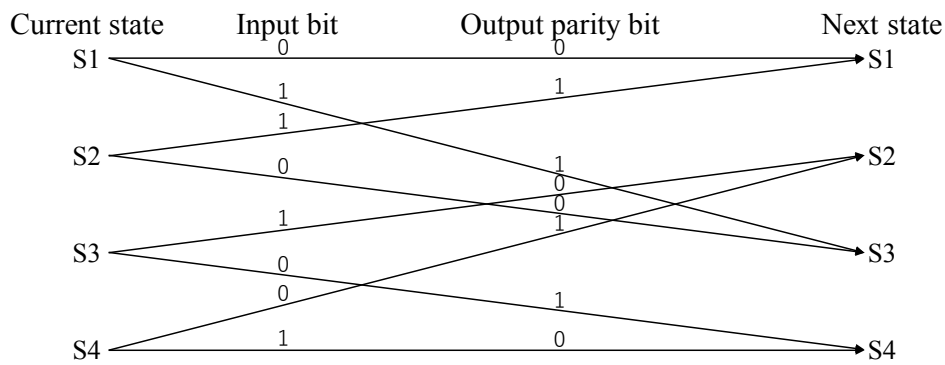


Figure 6-16 Trellis diagram of systematic convolutional code

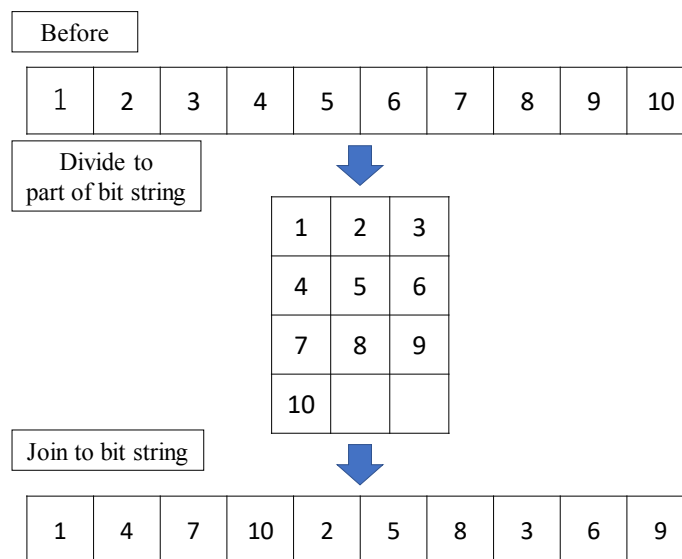


Figure 6-17 Interleaving process of implemented interleaver

Table 6-1 illustrates an example of an anonymized data table after the grouping process. The example table shows the degree of congestion of rooms classified as north, south, east, and west. In the table, records are sorted by “Time slot” item and then divided into two groups, G1 and G2. Table 6-2 is an example of an anonymized data table after the modification process when the watermarked information is ‘1011’ in binary format. In this example, “Area of room” is set to a target item of the modification and the modified values are underlined. In Table 6-2, G1 expresses the first bit from the left side of the watermarked information since the number of records in G1 is one. The value of the left most bit is ‘1’. One record in G1 has been modified because the value ‘1’ converted by the gray code is ‘1’. G2 can be expressed as any value from 0 to 7 by the watermarking method since it has 7 records. Therefore, G2 supports 3-bit bit strings. A part of the information that has not yet been watermarked is ‘011’ where the converted value of ‘011’ is ‘010’. Since the decimal format of ‘010’ is ‘2’, two of the records in G2 have been modified. The modification process degrades the value of the data due to the abstraction of the watermarking process. The modification process does not modify records repeatedly to minimize the degradation.

6 Watermarking method for anonymized data

Table 6-1 Example data table after grouping process

<u>Group</u>	<u>Time slot</u>	<u>Area of room</u>	<u>Congestion</u>
G1	8:00-8:30	East	Empty
	8:30-9:00	South	Full
	8:30-9:00	North	Medium
	8:30-9:00	East	Full
G2	8:30-9:00	West	Empty
	8:30-9:00	North	Medium
	8:30-9:00	South	Medium
	8:30-9:00	East	Full

Table 6-2 Example data table after modification process

<u>Group</u>	<u>Time slot</u>	<u>Area of room</u>	<u>Congestion</u>
G1	8:00-8:30	<u>South or East</u>	Empty
	8:30-9:00	South	Full
	8:30-9:00	North	Medium
	8:30-9:00	East	Full
G2	8:30-9:00	West	Empty
	8:30-9:00	North	Medium
	8:30-9:00	<u>South or East</u>	Medium
	8:30-9:00	<u>South or East</u>	Full

In this study, four abstraction methods were implemented for the proposed watermarking method to modify the values of the anonymized data while preventing any additional leaks of private information. The first method is named “masking method” which replaces a part of a value to a wildcard character ‘*.’ The second method is named “extension method” which adds candidates of a part of a value. For instance, “Hoge city” can be modified to “Hoge or Foo city” by the extension method. The third method is named “replacing method” which replaces a part of a value to a candidate of the value. This method is updated version of the replacing described in Section 2.1.4. It was updated to suppress the information loss to low levels. The last method is named “arranging method” which changes the order of a part of a value that has multiple candidates. Watermarked information can be extracted from watermarked anonymized data by comparing it with non-watermarked anonymized data, which is called the comparison process. In this process, the number of anonymized records used in the watermarking process are counted in each record group. The sequence of the number denotes the watermarked bit string that is encrypted by AES, coded by turbo code, and converted by gray code. The extraction flow described in Section 6.2.2 and Figure 6-15 show how the plaintext is extracted from the watermarked bit string. In contrast to general watermarking methods, the watermark of the proposed watermarking method is visible by comparison between watermarked and non-watermarked anonymized data since the proposed watermarking method modifies values. However, the proposed method can be categorized into watermarking techniques because data users are not able to notice the watermark without comparisons. Even if a data user finds differences by comparison between multiple watermarked data, the data user cannot extract the watermark from the statistical aspect as described in Section 6.2.1.

6 Watermarking method for anonymized data

The non-watermarked anonymized data is not published to the data users, making it impossible for the attackers to compare watermarked and non-watermarked anonymized data for watermark extraction. Therefore, the only way for the attackers to extract watermarked information is to compare multiple sets of published anonymized data. This is called collusion attack as mentioned in Section 6.2.1. The proposed watermarking method protects the watermarked information from collusion attacks by using the characteristic of the AES encryption as described in Section 6.2.1.

6.3 Specification of the implemented turbo code

6.3.1 Success rate of error correction

Before evaluating the proposed watermarking method, the error correction ability of the implemented turbo code was measured while shifting the length of a bit string input to the turbo code from 128 bits to 384 bits at 128-bit intervals. The input bit strings were the ciphertext of AES using the CTR mode, which was created by randomly generated plaintexts, secret keys, and nonces. Figure 6-18 illustrates the success rate of error corrections when some bits of input bit strings are flipped before the decoding process of the turbo code. The flipping rate against the input bit string was set from 1/256 to 1 while the decoding process was executed twenty times for each flipping rate. The internal decoder ran up to one thousand times for each of the decoding process. The iteration of the internal decoding process was a bottleneck in terms of the decoding processing time. To reduce the influence of the bottleneck, the iteration was aborted when the internal decoder output the same result for four times in a row.

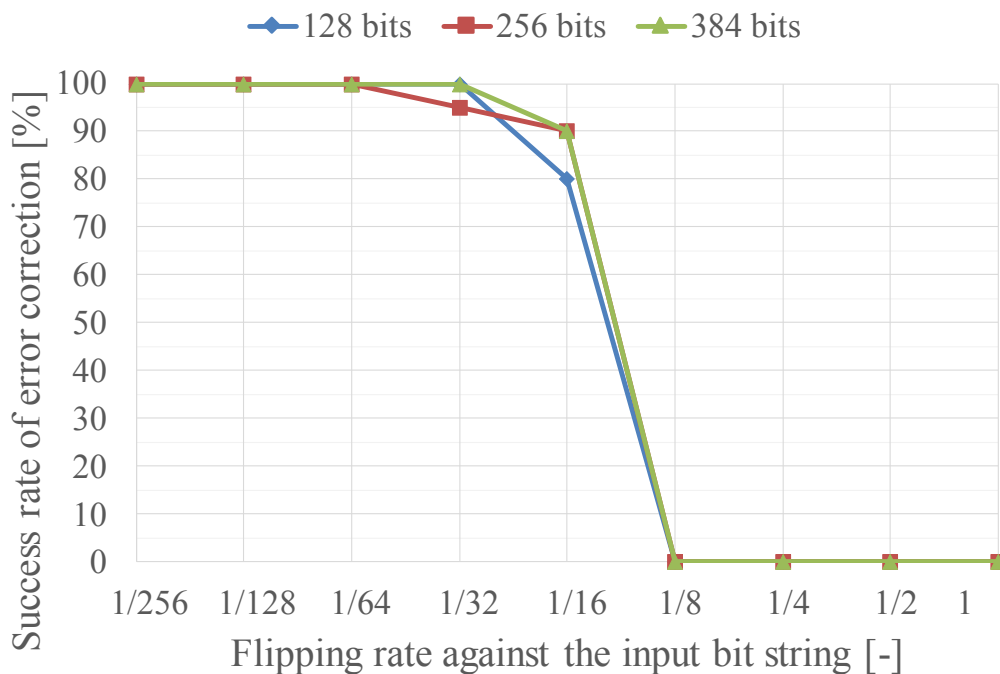


Figure 6-18 Success rate of error correction (1)

6 Watermarking method for anonymized data

Figure 6-18 shows that the success rate is greater than 80% when the flipping rate was 1/16 or smaller, and 0% when the flipping rate was 1/8 or larger. Figure 6-19 shows another result of the measurement with the same conditions as Figure 6-18 except for the flipping rate of the input bit string. The flipping rate was set from 1/16 to 1/8 to clarify the fluctuation of the success rate using these flipping rates. In Figure 6-19, the success rate fluctuated linearly. Also, for most of the flipping rates, the success rate was small compared to others where the input bit strings were longer than in this case. This is because SOVA calculates the most feasible path of state transition

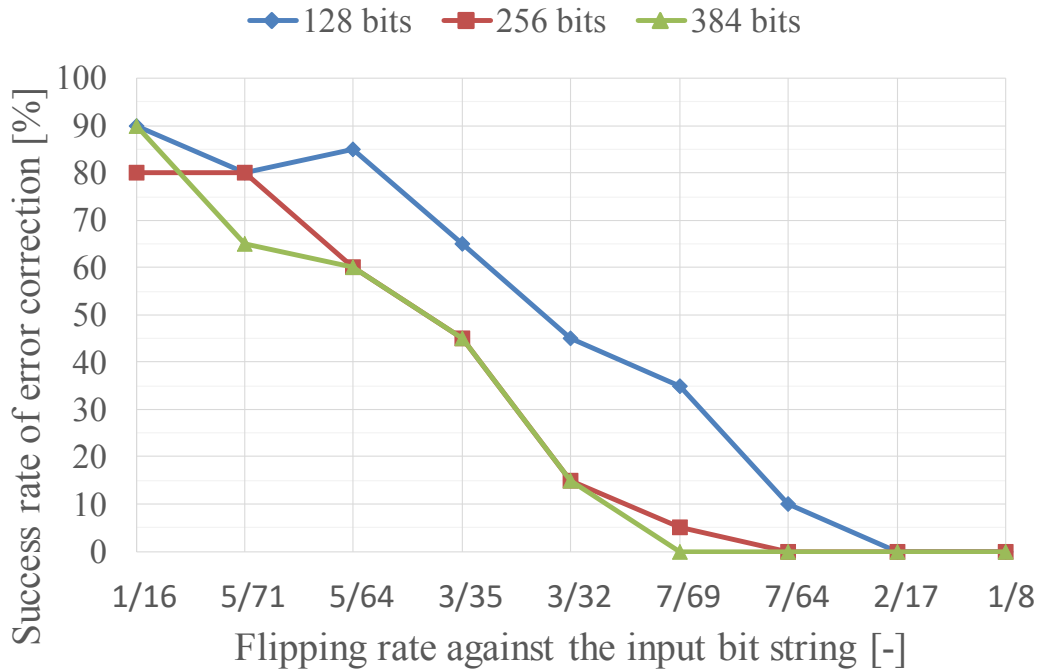


Figure 6-19 Success rate of error correction (2)

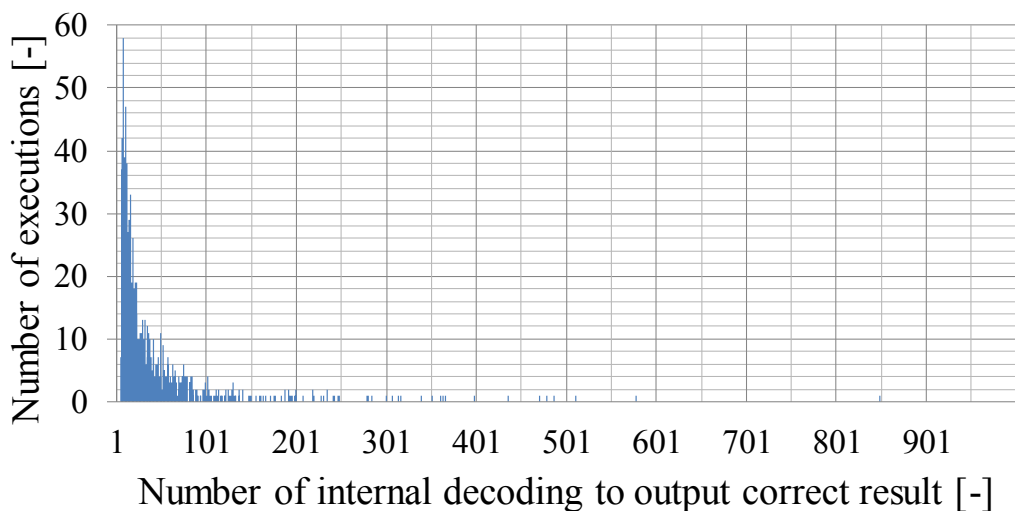


Figure 6-20 Distribution of number of decoding (128 bits)

6 Watermarking method for anonymized data

of the internal encoder at every 1 bit of the input bit string when decimating the paths to reduce the calculation cost of the decoding process. Therefore, the total number of the decimated paths is comparatively small when the length of the input bit string is short.

6.3.2 Validity of the maximum limit of iteration

The required number of iterations of the internal decoding was measured to output the correct decoding result when the length of the input bit string is 128, 256, and 384 bits to evaluate the validity of the maximum limit of iteration of the internal decoding process. The results are shown in Figure 6-20, Figure 6-21, and Figure 6-22, respectively. The number of flipped bits was set to 23, 42, and 64 bits, respectively, such that the success rate of the decoding process is 50% for each of the conditions. The numbers of flipped bits measured are displayed in Figure 6-19.

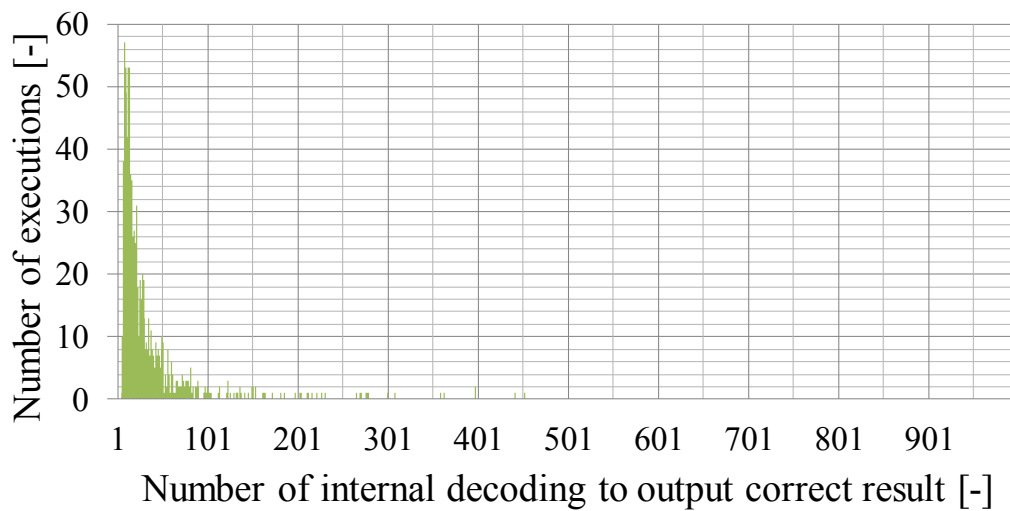


Figure 6-21 Distribution of number of decoding (256 bits)

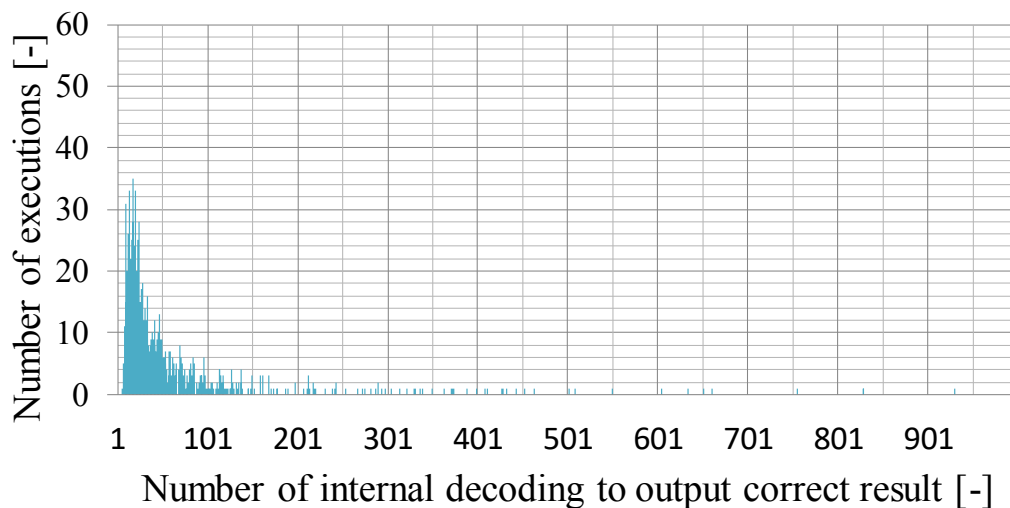


Figure 6-22 Distribution of number of decoding (384 bits)

6 Watermarking method for anonymized data

For each of the three figures, the horizontal axis indicates the number of iterations of the internal decoding executed until the result of the internal decoding became stable while the vertical axis indicates the number of the decoding processes. The decoding process was executed two thousand times for each of the three measurements. Executions that failed to decode were excluded since the evaluations focused on the influence of the maximum limit to the success rate. Executions that were unstable at the maximum limit of the iteration were also excluded as outliers. The averages of the number of the iterations required were 44.41, 33.88, and 61.31, respectively. The three measurement results show that most of the decoding processes became stable within four hundred iterations, and the required number of iterations to output the correct decoding result decreased exponentially when increasing the values of the vertical axis.

To confirm whether the maximum limit of one thousand is large enough or not, the rate of the executions of the internal decoding process that were iterated until the maximum limit was measured. As same as the measurement in Section 6.3.1, the flipping rate against the input bit string was set from $1/256$ to 1 while the decoding process was executed twenty times for each flipping rate. The rates were measured while the maximum limit was set to one thousand times, two thousand times, and three thousand times, respectively. The length of the input bit string was set to 128 bits. Table 6-3 shows the rate of executions that were unstable until the maximum limit while Figure 6-23 shows another measurement result that shows the success rate of the decoding process of the three patterns of the maximum limit. The rates in Table 6-3 get small when their maximum limit becomes large. However, according to Figure 6-23, there are no differences of the success rate among the three patterns. Therefore, the success rate of the internal decoding process is not improved even if the maximum limit is larger than one thousand times. This relationship between the success rate and the maximum limit were shown when the length of the input bit string was set to either 256 or 384 bits. For these results shown in Section 6.3.2, the maximum limit of the number of iterations is reasonable when the limit is one thousand.

6.3.3 Validity of aborting iterations of internal decoding

Table 6-3 Rate of executions that were unstable until the maximum limit

Maximum limit	Flipping rate against the input bit string								
	1/256	1/128	1/64	1/32	1/16	1/8	1/4	1/2	1
1,000 times	0%	0%	0%	0%	0%	75%	80%	90%	95%
2,000 times	0%	0%	0%	0%	0%	70%	65%	80%	95%
3,000 times	0%	0%	0%	0%	0%	70%	45%	65%	95%

Table 6-4 Rates of incorrect aborting

Aborting condition	Flipping rate of bits			
	1/16 +5/128	1/16 +6/128	1/16 +7/128	1/8
Four times	5%	0%	15%	0%
Six times	0%	0%	10%	0%
Eight times	0%	0%	5%	0%

6 Watermarking method for anonymized data

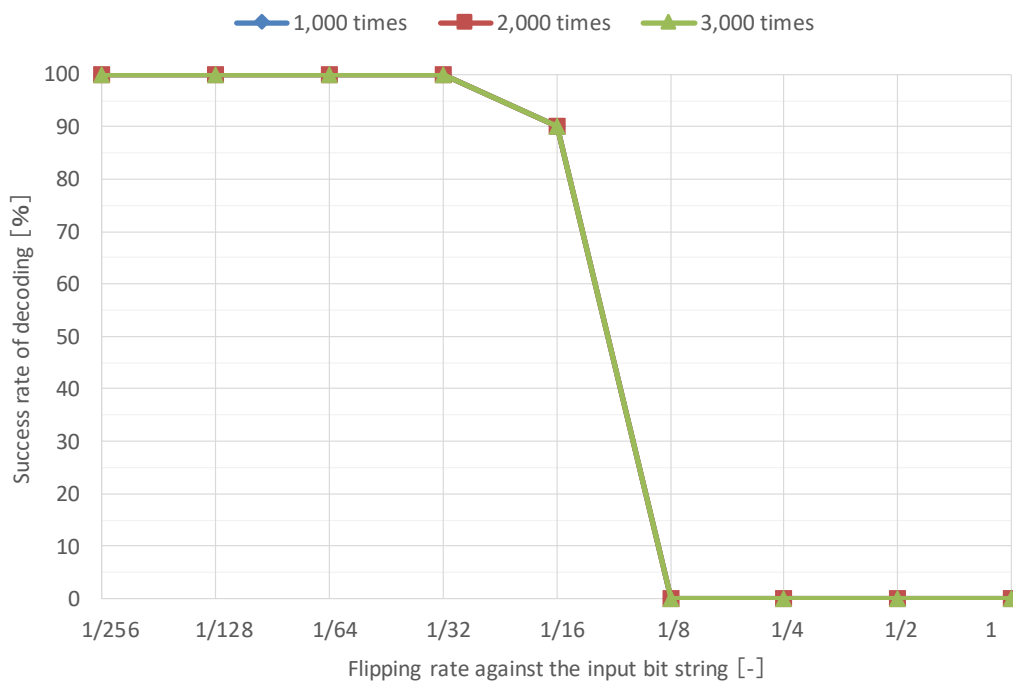


Figure 6-23 Success rate of the decoding process of the three patterns of the maximum limit (1,000, 2,000, and 3,000 times)

The implemented decoding process aborts the iteration of the internal decoding process when the internal decoder outputs the same result four times in a row. In this section, the rate where the decoded result of the aborted decoding process is different compared to the case when the internal decoding process is iterated one thousand times was measured. The rates were measured while shifting the flipping rate of bits of the input bit string from 1/16 to 1/8. Table 6-4 shows a part of the result of the measurement when the aborting condition is four, six, and eight times. The rates of all the aborting conditions were 0% when the flipping rate of bits was 1/16 + 4/128 or lower. According to Table 6-4, the rate is small when the aborting condition is large. When the decoded result at abortion was different from the result after thousand iterations, both results were incorrect. Therefore, the aborting condition of the implemented decoding process does not degrade the decoding process.

6.4 Evaluation

In this study, the proposed watermarking method was implemented. Additionally, three distortion attacks were executed using Python 3.5 to evaluate the proposed watermarking method tolerance to the distortion attacks. The distortion attacks are a deleting attack, an adding attack, and a replacing attack that respectively delete, add, or replace the records of the data. These are common types of distortion attacks in the watermarking domain.

6 Watermarking method for anonymized data

6.4.1 Tolerance against distortion attacks

The tolerance to the distortion attacks of the proposed watermarking method was evaluated by attempting to extract correct watermarked information from the anonymized data modified by the distortion attacks. Measurements for the evaluation were executed for each abstraction method. Anonymized data for the evaluations was generated from the location data of bus stops published at linkdata.org [81]. The location data of a bus stop itself does not contain private information. However, the data has to be anonymized if the data also indicates where passengers embarked and disembarked. Therefore, although the location data with information of passengers is valuable for secondary use such as for the marketing of shops along the bus line, attackers may be able to identify personal information of a specific passenger from the data. The number of records in the anonymized data was set to 3,000, which is the average number of passengers that ride on one bus in one month [82]. MeCab [83] was used to split the name of bus stops for anonymization.

The concordance rate of a watermarked bit string extracted from attacked anonymized data was measured when the bit string is compared with the correct bit string. Figure 6-24, Figure 6-25, and Figure 6-26 display the measurement results of the tolerance to the deleting, adding, and replacing attacks, respectively. The vertical axes indicate the average of the concordance rate while the measurement was executed twenty times for each attacking rate shown in the horizontal axes. The attacking rate expresses the rate of the number of records that are deleted, added, or replaced by the distortion attacks compared with the original number of records in the

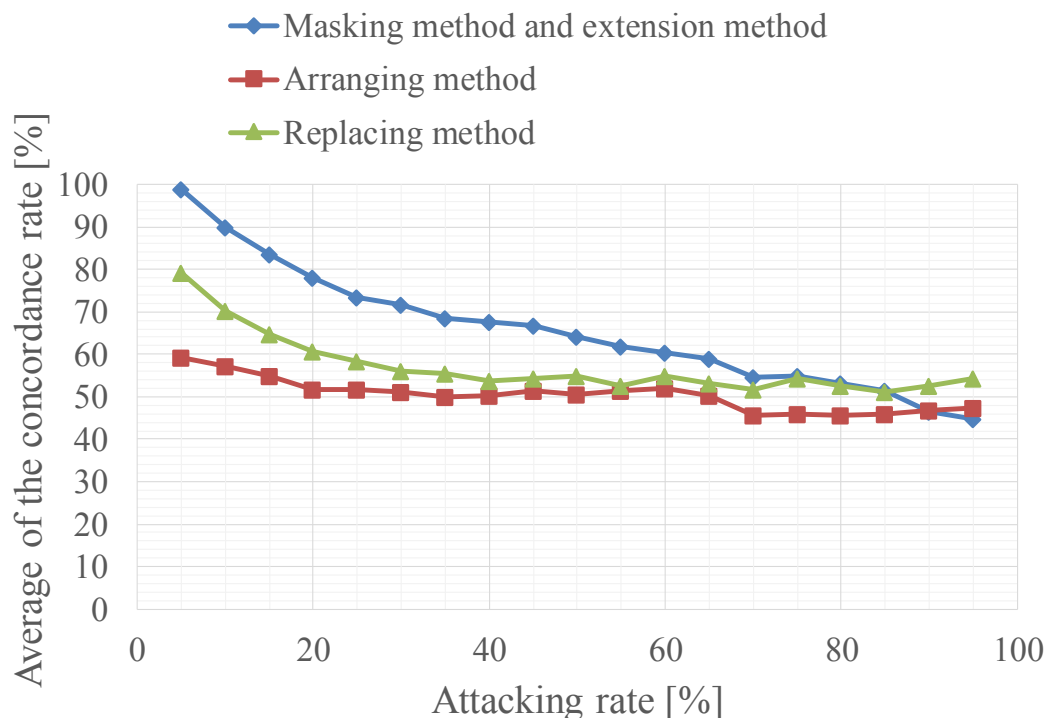


Figure 6-24 Tolerance against deleting attack

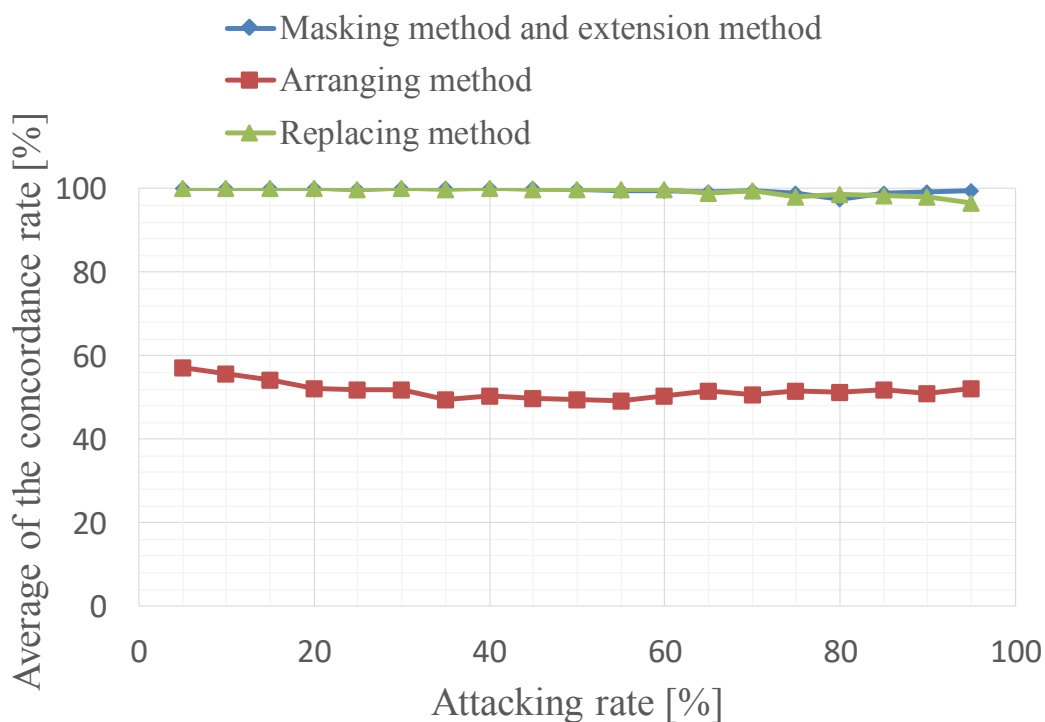


Figure 6-25 Tolerance against adding attack

data. The attacking rate was set from 5% to 95% at 5% intervals. Results of the masking method and the extension method were the same since their comparison algorithms at the extraction process of the watermarked information are the same. The watermarked information was 256 bits long.

Figure 6-24 shows that the most tolerant methods against the deleting attack are the masking method and the extension method when the attacking rate is 85% or smaller. Figure 6-25 shows that all abstraction methods except the arranging method keep the concordance rate to 96% or larger against the adding attack. Figure 6-26 shows similar measurement results as in Figure 6-24 because their algorithms of attack are similar. The replacing attack can be regarded as a combination of the deleting attack and the adding attack with the same attack rates, and the influence of the adding attack against this similarity is comparatively small since the proposed watermarking method is tolerant against the addition attack except when using the arranging method.

When the attacking rate is high, the extracted bit string consists of a large number of '0's since the proposed watermarking method expresses watermarked information by the number of modified records. Additionally, half of the ciphertext of AES consists of '0's. Therefore, the concordance rates in the three measurement results come close to 50%. The three measurement results show that the arranging method is not tolerant of distortion attacks since its concordance rate is 50%. However, this weakness can be an advantage when the arranging method is used to detect data modification.

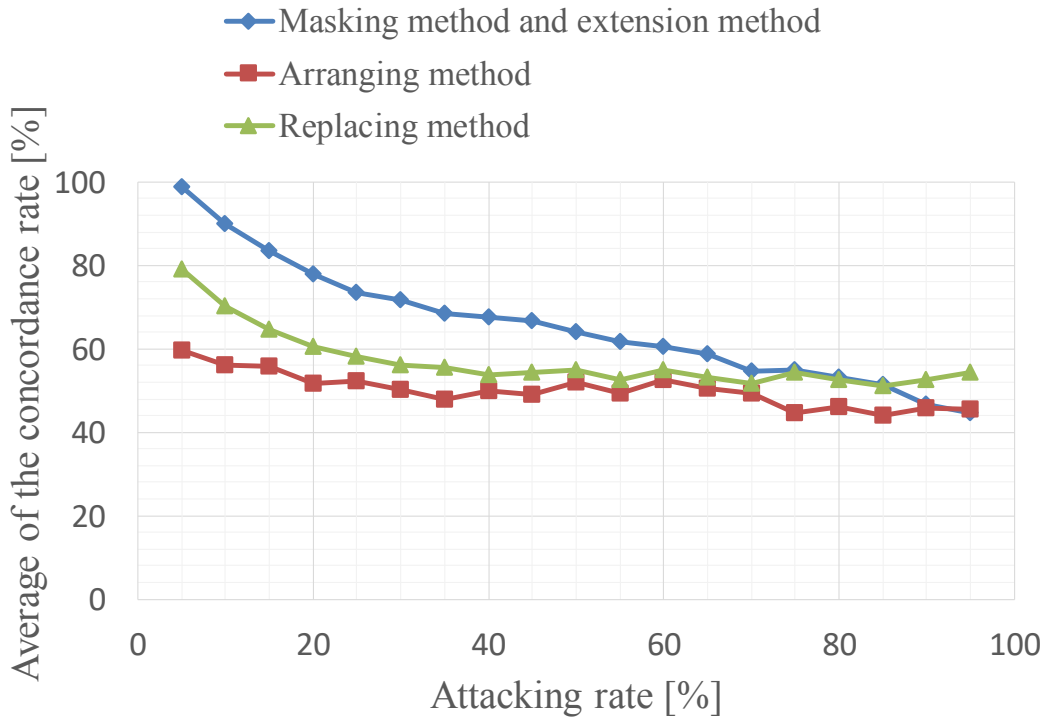


Figure 6-26 Tolerance against replacing attack

6.4.2 Effectiveness of use of gray code

The proposed watermarking method adopts to use the gray code to enhance the tolerance against distortion attacks. The success rate of error correction of two conditions that the implemented gray code was either enabled or disabled were compared to measure the effectiveness of the use of the gray code. The attacking rate was set from 5% to 95% at 5% intervals. Ten thousand records of the anonymized data used in Section 6.4.1 was used to the evaluation. The masking method and the deleting attack were selected to the abstraction method and the distortion attack, respectively. Figure 6-27 shows the result of the comparison. The vertical axes indicate the success rate of the error correction while the measurement was executed twenty times for each attacking rate shown in the horizontal axes. The result shows that the success rate is increased by using a gray code.

6.4.3 Availability of the proposed watermarking method in the proposed infrastructure

Situations that are similar to the attacks described in Section 6.4.1 can be supposed to exist where attackers use the same kind of distortion attacks in DTI4SFS. The adding attack can be regarded as a situation where multiple anonymized data tables are gathered into one data table. According to the measurement results, the proposed watermarking method can extract at least 96% of the watermarked information even if the anonymized data is combined with other data

6 Watermarking method for anonymized data

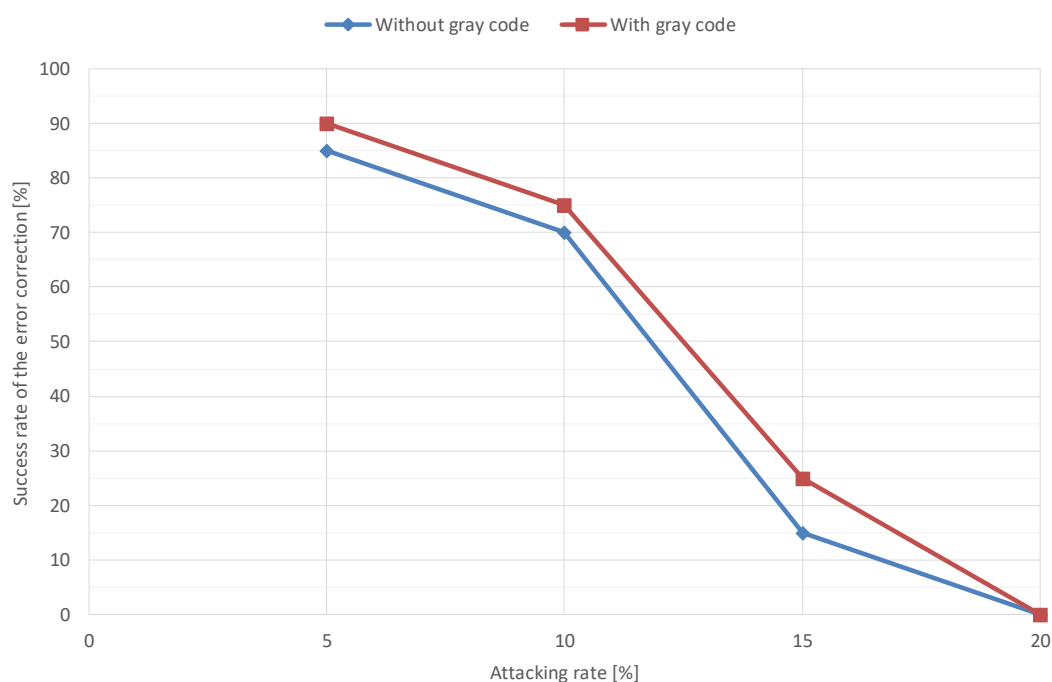


Figure 6-27 Comparison of success rates of the error correction between using and not using gray code


whose size is 95% of the anonymized data. The deleting attack can be regarded as a part of the anonymized data being republished. In the deleting attack, the concordance rates of the masking method and the extension attack are 50% when the attacking rate is 70% or higher. Therefore, the proposed watermarking method is effective in identifying whether the watermarking information is embedded or not if the republished data includes at least 30% of the original anonymized data. This tolerance is sufficient because anonymized data has no value when 70% of its records are lost.

6.4.4 Information loss due to watermarking

The proposed watermarking method modifies values of anonymized data using the four abstraction methods described in Section 6.2.3. Since the abstraction methods except the arranging method generalize the values, these three abstraction methods cause information loss. Figure 6-28 shows information loss in the evaluation of Section 6.4.1. The extension method and replacing method abstract a part of a value by adding candidates of the part or replacing the part whereas the masking method completely hide the part. Because of the difference, the information of both the extension and replacing methods are 19 times smaller than the masking method. Since the arranging method does not modify the values, this abstraction method can be used to detect data modification without information loss caused by the arranging method.

6 Watermarking method for anonymized data

Abstraction methods	Increase of information loss
Masking method	0.1508
Replacing method	0.007829
Extension method	0.007829
Arranging method	0



**19 times smaller than
masking method**

Figure 6–28 Comparison of Information loss among abstraction methods

6.5 Summary

In this section, a watermarking method for anonymized data was proposed and implemented while assuming the proposed method is used to identify the user that republished the anonymized data without authorization within DTI4SFS. The proposal facilitates the secondary use of data that may include privacy information by providing an approach to generate identifiable anonymized data. The proposed watermarking method consists of AES, turbo code, and a gray code converter to protect watermarking information from collusion attacks and distortion attacks. Experimental results showed that the proposed watermarking method could extract more than 95% of the watermarked information from anonymized data even if the data is combined with other records whose size is 95% of the data. Also, the embedding of watermarked information can be detected from data that is 30% of the anonymized data. For the information loss caused by the proposed abstraction methods, the experimental result showed that the extension and replacing methods were 19 times smaller than the masking method.

7 Conclusion

In this study, a data transaction infrastructure for safe and flexible sharing of private information (DTI4SFS) has been proposed. DTI4SFS uses anonymization techniques for the safety while focusing on data sharing. Anonymization generalizes values in data to eliminate uniqueness of records in the data. The generalization prevents identification of a record relating to a specific person by attackers. For the flexibility, DTI4SFS enables to share data including containing private information from data providers to data users while preserving regarding data privacy. DTI4SFS allows both data providers and data users to claim their requirements regarding the data sharing by creating either publishing or request rules, which are based on a proposed format named XML-based Anonymization Sheets (XAS). The detail of DTI4SFS have been proposed and described in Section 3.

Several methods have also been proposed in this study to enhance abilities of DTI4SFS. One of the proposed methods is regarding the anonymization process, which is a core process in DTI4SFS. The anonymization process tends to be a bottleneck of the entire process of DTI4SFS because all the data must be anonymized. For overcoming the bottleneck, a hardware implementation of an anonymizer has been proposed in Section 4. The proposed anonymizer showed 82.3% reduction in circuit size compared to existing anonymizer based on TCAM. The maximum throughput of the anonymizer was 8.75Gbps. The throughput of the proposed anonymizer is enough to anonymize data at a line speed of OC48 or faster until 8.75Gbps.

When focusing on data type, it is better if DTI4SFS can share time-series data because some data such as sensor data take this data type. On the other hand, general anonymization methods are not suitable for time-series data. In Section 5, an anonymization method using self-organizing map to share time-series data has been proposed so that DTI4SFS can share time-series data while preserving data privacy using anonymization. The proposed anonymization method achieves the same level of k -anonymity with small information loss compared with other conventional method that each data provider anonymizes aggregated data without data sharing among data providers. Information loss was reduced up to 22%.

In this study, a watermarking method for anonymized data has been proposed to suppress unauthorized republishing from malicious data users. Anonymized data published from DTI4SFS are same when requests from data users are same each other. Therefore, the malicious data user is not able to be identified from the republished anonymized data. For overcoming the problem, the watermarking method makes the anonymized data unique for each data user even if multiple data users submit the same request rules. The watermarking method adds a watermark into the data by modifying the values of the anonymized data to identify the users who received the data from DTI4SFS. The proposed watermarking method consists of AES, turbo code, and a gray code converter to protect watermarking information from collusion attacks and distortion attacks. Experimental results showed that the proposed watermarking method could extract more than

7 Conclusion

95% of the watermarked information from anonymized data even if the data is combined with other records whose size is 95% of the data. Additionally, the embedding of watermarked information can be detected from data that is 30% of the anonymized data. For the information loss caused by the proposed abstraction methods, the experimental result showed that the extension and replacing methods were 19 times smaller than the masking method.

The data transaction infrastructure proposed in this study enables data sharing flexibly while preserving private information contained in the shared data. The proposed infrastructure facilitates the secondary use of data including private information since the proposed infrastructure enables data sharing among third-party data users. If big data are shared for secondary use through the proposed infrastructure, applications that have not realized yet would be developed so that the applications make our lives better while preserving our privacy.

DTI4SFS facilitates services with secondary use of data because DTI4SFS allows data providers, i.e. the clients of the services, to manage and publish their data while preserving their privacy. Additionally, DTI4SFS enables data providers to actively manage their data for data publishing. This feature would promote secondary use of data by reducing anxiety of data providers regarding the data privacy.

For realizing the services, further practical studies would be required such as the implementation of DTI4SFS while assuming concrete situations and services. The author of this dissertation has studied regarding secure data communication on wireless sensor network [84]. This study will be applicable to implement DTI4SFS on IoT networks. Here is another study regarding precise time synchronization [85]. Although this study would not be directly applicable for the implementations, the idea of the study would be useful to provide strict data transaction in terms of precise timestamp. Further studies to increase number of data types for data sharing would also be required such as location data to enhance scalability of DTI4SFS. Moreover, from the view point of usability, studies to propose applications that use anonymized data based on DTI4SFS are meaningful. Through the study in this dissertation and these future works, future that people share their data with privacy preservation would be realized.

References

- [1] A. Perrin, M. Duggan, L. Rainie, A. Smith, S. Greenwood, M. Porteus , D. Page, “Social Media Usage: 2005-2015,” Pew Research Center, <http://www.pewinternet.org/2015/10/08/social-networking-usage-2005-2015/>, 2015.
- [2] L. Abraham, J. Allen, O. Barykin, V. Borkar, B. Chopra, C. Gereia, D. Merl, J. Metzler, D. Reiss, S. Subramanian, J. Wiener , O. Zed, “Scuba: Diving into Data at Facebook,” the 39th International Conference on Very Large Data Bases (VLDB), vol.6, no.11, 2013.
- [3] D. Reinsel, J. Gantz and J. Rydning, "The Digitization of the World - From Edge to Core," <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>, IDC, 2018.
- [4] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah , P. Siano, “Iot-based smart cities: A survey,” 016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), pp. 1-6, 2016.
- [5] L. D. Xu, W. He , S. Li, “Internet of Things in Industries: A Survey,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, 2014.
- [6] S. M. R. Islam, M. H. K. D. Kwak, M. Hossain , K. S. Kwak, “The Internet of Things for Health Care: A Comprehensive Survey,” in IEEE Access, vol.3, pp.678-708, 2015.
- [7] Cabinet office of government of Japan, “Society 5.0,” http://www8.cao.go.jp/cstp/english/society5_0/index.html, accessed August 15th, 2018.
- [8] The Open Definition, “The Open Definition,” <https://opendefinition.org/>, accessed November 16th, 2018.
- [9] data.gov, “data.gov,” <https://www.data.gov/>, accessed November 6th, 2018.
- [10] data go.jp, “data go.jp,” <http://www.data.go.jp/>, accessed November 6th, 2018.
- [11] European Parliament and Council of the European Union, “Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data,” <http://data.europa.eu/eli/dir/1995/46/oj>, accessed November 6th, 2018, 1995.
- [12] European Commission, “EU-US Privacy Shield,” https://ec.europa.eu/info/law/law-topic/data-protection/data-transfers-outside-eu/eu-us-privacy-shield_en, accessed August 15th, 2018.
- [13] European Commission, “2018 reform of EU data protection rules,” https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en, accessed August 15th, 2018.
- [14] “Personal Information Protection Commission, Laws and Policies,” <https://www.ppc.go.jp/en/legal/> , accessed August 16th, 2018.
- [15] J. Sakuma , S. Kobayashi, “Privacy-Preserving Data Mining,” Journal of Japanese Society

- for Artificial Intelligence, vol.24, no.2, pp.283-294, 2009.
- [16] R. Agrawal , R. Srikant, “Privacy-preserving data mining,” ACM Sigmod Record, vol.29, pp.439-450, 2000.
- [17] Y. Lindell , B. Pinkas, “Privacy preserving data mining,” Journal of cryptology, vol.15, no.3, pp.177-206, 2002.
- [18] J. Vaidya, C. W. Clifton , M. Y. Zhu, Privacy preserving data mining, Springer Science & Business Media, vol.19, 2006.
- [19] DIMACS Center, CoRE Building, Rutgers University, Piscataway, NJ, “DIMACS/PORTIA Working Group Meeting on Privacy-Preserving Data Mining,” <http://dimacs.rutgers.edu/Workshops/WGDatasets/abstracts.html>, accessed November 6th, 2018, 2004.
- [20] M. Naehrig, K. Lauter , V. Vaikuntanathan, “Can homomorphic encryption be practical?,” Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW), pp.113-124, 2011.
- [21] C. Gentry, “Fully homomorphic encryption using ideal lattices,” In STOC, pp.169-178, 2009.
- [22] J. Zhou, Z. Cao, X. Dong , X. Lin, “PPDM: A privacy-preserving protocol for cloud-assisted e-healthcare systems,” in IEEE Journal of Selected Topics in Signal Processing, vol.9, no.7, pp.1332-1344, 2015.
- [23] B. C. M. Fung, K. Wang, R. Chen , P. S. Yu, “Privacy-preserving data publishing: A survey of recent developments,” ACM Computing Surveys (CSUR), vol.42, no.4, 2010.
- [24] L. Sweeney, “k-anonymity: a model for protecting privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol.10, no.5, pp.557-570, 2002.
- [25] L. Sweeney, “Achieving k-anonymity privacy protection using generalization and suppression,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol.10, no.5, pp.571-588, 2002.
- [26] A. Machanavajjhala, D. Kifer, J. Gehrke , M. Venkatasubramanian, “l-diversity: Privacy beyond k-anonymity,” ACM Transactions on Knowledge Discovery from Data (TKDD), vol.1, no.1, 2007.
- [27] N. Li, T. Li , S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” IEEE 23rd International Conference on Data Engineering (ICDE), pp.106-115, 2007.
- [28] B.-C. Chen, D. Kifer, K. LeFevre , A. Machanavajjhala, “Privacy-Preserving Data Publishing,” Foundations and Trends in Databases, vol.2, no.1-2, pp.1-167, 2009.
- [29] C. C. Aggarwal, “On Randomization, Public Information and the Curse of Dimensionality,” IEEE 23rd International Conference on Data Engineering (ICDE), pp.136-145, 2007.
- [30] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi , A. W. Fu, “Utility-based anonymization using local recoding,” Proceedings of the 12th ACM international conference on Knowledge discovery and data mining (SIGKDD), pp.785-790, 2006.
- [31] R. J. Bayardo , R. Agrawal, “Data privacy through optimal k-anonymization,” 21st International Conference on Data Engineering (ICDE), pp.217-228, 2005.
- [32] A. Asayesh, M. A. Hadavi , R. Jalili, “(t,k)-Hypergraph anonymization: an approach for

- secure data publishing,” *Journal of Security and Communication Networks*, vol.8, no.7, pp.1306-1317, 2015.
- [33] K. Shared , G. Danezis, “An Automated Social Graph De-anonymization Technique,” *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES)*, pp.47-58, 2014.
- [34] X. Zhang, L. T. Yang, C. Liu , J. Chen, “A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol.25, no.2, pp.363-373, 2014.
- [35] S. Gambs, M. O. Killijian , M. N. P. Cortez, “De-anonymization attack on geolocated data,” *Journal of Computer and System Sciences*, vol.80, no.8, pp.1597-1614, 2014.
- [36] A. Narayanan , V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” *IEEE Symposium on Security and Privacy*, pp.111-125, 2008.
- [37] A. Øhrn , L. Ohno-Machado, “Using Boolean reasoning to anonymize databases,” *Artificial Intelligence in Medicine*, Vol.15, no.3, pp.235-254, 1999.
- [38] Y. Rubner, C. Tomasi , L. J. Guibas, “The Earth Mover's Distance as a Metric for Image Retrieval,” *International Journal of Computer Vision*, vol.40, no.2, pp.99-121, 2000.
- [39] E. A. McGlynn, T. A. Lieu, M. L. Durham, A. Bauck, R. Laws, A. S. Go, J. Chen, H. S. Feigelson, D. A. Corley, D. R. Young, A. F. Nelson, A. J. Davidson, L. S. Morales , M. G. Kahn, “Developing a data infrastructure for a learning health system: the PORTAL network,” *Journal of the American Medical Informatics Association*, vol.21, no.4 pp.596-601, 2014.
- [40] D. A. Corley, H. S. Feigelson, T. A. Lieu , E. A. McGlynn, “Building Data Infrastructure to Evaluate and Improve Quality: PCORnet,” *Journal of Oncology Practice*, vol.11, no.3, pp.204-206, 2015.
- [41] L. N. Shulman, R. McCabe, G. Gay, B. Palis , D. McKellar, “Building Data Infrastructure to Evaluate and Improve Quality: The National Cancer Data Base and the Commission on Cancer's Quality Improvement Programs,” *Journal of Oncology Practice*, vol.11, no.3, pp.209-212, 2015.
- [42] M. Balamurugan, J. Bhuvana , S. ChenthurPandian, “Shared and secured data partitioning for privacy preserving of collaborative file transfer in multi path computational mining,” *2012 Ninth International Conference on Wireless and Optical Communications Networks (WOCN)*, pp.1-7, 2012.
- [43] R. K. Abeysekara , W. Zhang, “Hybrid framework for privacy preserving data sharing,” *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp.198-206, 2013.
- [44] C. J. L. Xu, J. Wang, J. Yuan , Y. Ren, “Information Security in Big Data: Privacy and Data Mining,” in *IEEE Access*, vol.2, pp.1149-1176, 2014.
- [45] P.-C. Lin , Y.-W. Lin, “Towards packet anonymization by automatically inferring sensitive application fields,” *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pp.87-92, 2012.
- [46] L. Shou, X. Shang, K. Chen, G. Chen , C. Zhang, “Supporting Pattern-Preserving Anonymization for Time-Series Data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol.25, No.4, pp.877-892, 2013.
- [47] Z. K. Baker , V. K. Prasanna, “An Architecture for Efficient Hardware Data Mining using

- Reconfigurable Computing Systems,” 2006 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, CA pp.67-75, 2006.
- [48] Y.-H. Wen, J.-W. Huang , M.-S. Chen, “Hardware-Enhanced Association Rule Mining with Hashing and Pipelining,” in IEEE Transactions on Knowledge and Data Engineering, vol.20, no.6, pp.784-795, 2008.
- [49] J. Sawada , H. Nishi, “Hardware accelerator for low latency privacy preserving mechanism,” Proceeding in 4th International Conference on Future Computational Technologies and Applications, FUTURE COMPUTING 12, 2012.
- [50] J. Sawada , H. Nishi, “Hardware acceleration and data-utility improvement for low-latency privacy preserving mechanism,” 22nd International Conference on Field Programmable Logic and Applications (FPL), pp.499-502, 2012.
- [51] F. Yamaguchi , H. Nishi, “Hardware-based Hash Functions for Network Applications,” 2013 19th IEEE International Conference on Networks (ICON), pp.1-6, 2013.
- [52] W. W. Peterson , D. T. Brown, “Cyclic Codes for Error Detection,” Proceedings of the IRE, pp.228-235, 1961.
- [53] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” Communications of the ACM, vol.13, no.7, pp.422-426, 1970.
- [54] E. L. Quinn, “Smart Metering and Privacy: Existing Laws and Competing Policies,” Colorado Public Utilities Commission, 2009.
- [55] O. Parson, S. Ghosh, M. Weal , A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” at Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI-12), pp.356-362, 2012.
- [56] M. Figueiredo, A. D. Almeida , B. Ribeiro, “Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems,” Neurocomputing, vol.96, pp.66-73, 2012.
- [57] V. Y. Pillitteri , T. L. Brewer, “Guidelines for Smart Grid Cybersecurity,” NIST Interagency/Internal Report (NISTIR) 7628 Rev.1, 2014.
- [58] C. Neureiter, G. Eibl, A. Veichtlbauer , D. Engel, “Towards a framework for engineering smart-grid-specific privacy requirements,” 39th Annual Conference of the IEEE Industrial Electronics Society, pp.4803-4808, 2013.
- [59] T. Kohonen, “Self-organized formation of topologically correct feature maps,” Biological Cybernetics, vol.43, no.1, pp.59-69, 1982.
- [60] Z. Guan, G. Si, X. Du, P. Liu, Z. Zhang , Z. Zhou, “Protecting user privacy based on secret sharing with fault tolerance for big data in smart grid,” IEEE International Conference on Communications (ICC), pp.1-6, 2017.
- [61] S. C. a. K. G. Srinivasagan, “A combined random noise perturbation approach for multi-level privacy preservation in data mining,” 2014 International Conference on Recent Trends in Information Technology, pp.1-6, 2014.
- [62] K. Kursawe, G. Danezis , M. Kohlweiss, “Privacy-friendly aggregation for the smart-grid,” International Symposium on Privacy Enhancing Technologies Symposium (PETS), pp.175-191, 2011.
- [63] J.-W. Byun, A. Kamra, E. Bertino , N. Li, “Efficient k-anonymization using clustering techniques,” Proceedings of the 12th international conference on Database systems for advanced applications, pp.188-200, 2007.

- [64] T. W. Liao, "Clustering of time series data—a survey," *Journal of Pattern Recognition*, vol.38, no.11, pp.1857-1874, 2005.
- [65] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.7, pp.881-892, 2002.
- [66] K. Okada, K. Matsui, J. Haase, H. Nishi, "Privacy-preserving data collection for demand response using self-organizing map," *IEEE 13th International Conference on Industrial Informatics (INDIN)*, pp.652-657, 2015.
- [67] G. E. P. Box, G. M. Jenkins, "Time series analysis: forecasting and control," San Francisco: Holden-Day, 1976.
- [68] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *EUROCRYPT'99 Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, pp.223-238, 1999.
- [69] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol.31, no.4, pp.469-472, 1985.
- [70] H. Akaike, "A new look at the statistical model identification," in *IEEE Transactions on Automatic Control*, vol.19, no.6, pp.716-723, 1974.
- [71] I. S. S. D. Archive, "Data from the commission for energy regulation," <http://www.ucd.ie/issda/data/commissionforenergyregulationcer/>, accessed November 6th, 2018.
- [72] J. S. A. Johnsana, A. Rajesh, S. K. Verma, "CATs-clustered k-anonymization of time series data with minimal information loss and optimal re-identification risk," *Indian Journal of Science and Technology*, vol.9, no.47, pp.1-13, 2016.
- [73] V. Rajalakshmi, G. S. A. Mala, "Anonymization by data relocation using sub-clustering for privacy preserving data mining," *Indian Journal of Science and Technology*, vol.7, no.7, pp.975-980, 2014.
- [74] C. I. Podilchuk, E. J. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine*, vol.18, no.4, pp.33-46, 2001.
- [75] P. Singh, R. S. Chadha, "A survey of digital watermarking techniques, applications and attacks," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol.2, no.9, pp.165-175, 2013.
- [76] H. Yuki, *Introductory technology of cryptography*, SB Creative, 2008 (in Japanese).
- [77] N. Ferguson, B. Schneier, *Practical Cryptography*, Wiley, 2003.
- [78] H. Matsuoka, K. Inoue, H. Nishi, "Perfect Classified Channel retaining DC balance," *IEICE technical report*, vol.108, no.15, DC2008-1, pp.1-6, 2008 (in Japanese).
- [79] Y. Nishimura, *Bases of data coding techniques and error correction*, CQ Publishing, 2010 (in Japanese).
- [80] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM)*, vol.3, pp.1680-1686, 1989.
- [81] LinkData, "Link and Publish your data | Open data sharing," <http://linkdata.org>, accessed November 6th, 2018.

- [82] Ministry of Land, Infrastructure, Transport and Tourism, “Statistical Survey of Automobile Transportation,” <http://www.mlit.go.jp/k-toukei/06/annual/index.pdf>, accessed November 6th, 2018, 2015 (in Japanese).
- [83] T. Kudo, “MeCab: Yet Another Part-of-Speech and Morphological Analyzer,” <http://taku910.github.io/mecab/>, 2013 (in Japanese).
- [84] Y. Nakamura, M. Louvel , H. Nishi, “Coordination middleware for secure wireless sensor networks,” 42nd Annual Industrial Electronics Conference (IECON2016), 2016.
- [85] Y. Nakamura, A. Harvath , H. Nishi, “Time Synchronization Technique Using EPON for Next-Generation Power Grids,” IEICE Trans. on Communications, Vol.E99-B, No.4, pp.859-866, 2016.

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Professor Hiroaki Nishi for providing me this precious study opportunity as a Ph.D student in his laboratory. I am deeply grateful to Professor Takamichi Saito in Meiji University, Professor Kenji Kono and Associate Professor Ryogo Kubo in Keio University for their valuable comments, discussions and feedbacks for this study. I would like to thank all Nishi laboratory members, especially Mrs. Yuko Nishi, Toshiro Togoshi, Kouichi Inoue, Taisei Hayashi, Shinichi Ishida, Hayato Yamaki, Janaka Wijekoon, Rajitha Tennekoon, Shanaka Prageeth. Erwin Harahap, Kazuki Masuda, Kazumasa Ikeuchi, Daigo Hogawa, Fumito Yamaguchi, Satoshi Koibuchi, Kenichi Takagiwa, Kengo Okada, Hironori Okano, Mio Fukuta, Shoki Kawano, and Sachio Godo. I received generous support and help from all the members. I would also like to take this opportunity to thank all the members of the laboratory that I joined in CEA Grenoble for their hospitality and training for my research skills. I owe my profound gratitude to Emeritus Professor Takashi Nodera and the members of Information Technology Center in Keio University for their understanding and generous support. Finally, I would like to offer my special thanks to my parents for their moral support and warm encouragement.

Yuichi NAKAMURA
February 2019