Journal of Korea Multimedia Society Vol. 20, No. 11, November 2017(pp. 1785-1792)
https://doi.org/10.9717/kmms.2017.20.11.1785

하둡 분산 환경 기반 프라이버시 보호 빅 데이터 배포 시스템 개발

김대호†, 김종욱††

Development of a Privacy-Preserving Big Data Publishing System in Hadoop Distributed Computing Environments

Dae-Ho Kim†, Jong Wook Kim††

ABSTRACT

Generally, big data contains sensitive information about individuals, and thus directly releasing it for public use may violate existing privacy requirements. Therefore, privacy-preserving data publishing (PPDP) has been actively researched to share big data containing personal information for public use, while protecting the privacy of individuals with minimal data modification. Recently, with increasing demand for big data sharing in various area, there is also a growing interest in the development of software which supports a privacy-preserving data publishing. Thus, in this paper, we develops the system which aims to effectively and efficiently support privacy-preserving data publishing. In particular, the system developed in this paper enables data owners to select the appropriate anonymization level by providing them the information loss matrix. Furthermore, the developed system is able to achieve a high performance in data anonymization by using distributed Hadoop clusters.

Key words: Privacy-Preserving Data Publishing, Hadoop, Distributed Computing, k-Anonymity,

1. 서 론

모바일 기기의 대중화와 사물인터넷 시대의 등장으로 인하여, 많은 분야에서 빅데이터가 폭발적으로 생성되고 있다. 빅데이터는 미래 신성장 동력이자 신산업으로서, 데이터 개방과 공유 패러다임의 변화와 함께 산업, 경제, 사회 전반에서 크게 활용될 것으로 기대되고 있다. 최근 수년간 빅데이터 분석이 큰 가치를 창출할 것으로 주목받아 왔으며, 이로 인하여 관련 기술 또한 활발히 개발되고 있다. 또한, 이에 발 맞춰, 정부는 교통, 인구, 환경 등 다양한 공공 빅데이터를 민간에 공유하고 있으며, 일반인들의 활발한 빅데이터 활용을 가능하게 하여 보다 다양한 가치 창출을 촉진하고 있다.

하지만 빅데이터의 공유가 증가함에 따라 개인정보 유출의 위험성도 함께 증가하고 있다. 다양한 환경에서 생산되는 데이터는 수많은 정보들을 포함하고 있으며, 그 중 개인을 식별할 수 있는 데이터들도 포함되어 있기 때문이다. 이러한 개인의 민감한 정보는 사생활 침해 등 심각한 사회문제를 야기할 수 있

※ Corresponding Author : Jong Wook Kim, Address: (110-743) 20-Gil, Hongji-dong, Jongno-gu, Seoul, Korea, TEL : +82-2-781-7590, FAX : +82-2-781-7590, E-mail : jkim@smu.ac.kr
Receipt date : Sep. 4, 2017, Revision date : Oct. 17, 2017
Approval date : Oct. 18, 2017

† Dept. of Computer Science, Sangmyung University (E-mail :rlaeogh222@gmail.com)
†† Dept. of Computer Science, Sangmyung University
※ This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2017-0-01703, Wireless Transmission System for Full-Parallax Multiview)

다. 그러므로 데이터 공유에 있어 개인의 민감한 데이터들은 제거하거나 비식별화함으로써 개인정보 유출을 사전에 방지해야 한다.

최근 들어 데이터 배포에서 최소한의 데이터 변형으로 개인의 프라이버시를 보호하는 프라이버시 보호 데이터 배포(Privacy-Preserving Data Publishing, PPDP)가 활발히 연구되어 왔다[1,2,3]. 프라이버시 보호 데이터 배포는 프라이버시 모델에 따라 원본 데이터를 변형하는 방식이 다르며, 현재까지 다양한 프라이버시 모델이 연구되어 왔다. (예, k-익명성 [4,5,6,7,8,9,10], l-다양성 [11], t-근접성 [12]). 그 중 가장 대표적인 k-익명성은 적어도 k개의 데이터가 준식별자 속성에 대해 같은 값을 가지도록 데이터를 변형하여 프라이버시를 보호하는 방법이다[4,5,6,7,8,9,10]. 또한, l-다양성은 k-익명성을 보장함과 동시에 데이터 집합에서 l개의 민감한 정보를 분포시켜 데이터를 비식별화 하는 방법이다[11]. t-근접성은 k-익명성과 l-다양성의 취약점을 보완하고자 제안된 모델로서, 민감한 정보가 특정 값으로 쏠리거나, 유사한 값으로 뭉치는 경우를 방지하기 위하여 민감한 정보의 분포(t)를 조정하여 개인정보를 보호하는 기법이다 [12]. 현재 이와 같은 다양한 비식별화 기술을 활용함으로써 빅 데이터의 공유시 발생할 수 있는 개인 정보 유출을 방지하기 위한 노력을 기울이고 있다.

다양한 분야에서 빅데이터 공유에 대한 요구가 증가함에 따라, 이를 지원하기 위한 시스템 개발에 관한 관심도 높아지고 있다[17]. 그러므로 본 논문에서는 프라이버시 보호 빅데이터 배포를 지원하는 시스템을 개발한다. 특히, 본 논문에서 개발한 시스템은 데이터 소유자가 빅데이터를 배포함에 있어 원하는 데이터를 k-익명화하여 제공함으로써 데이터 배포에서 발생할 수 있는 개인정보 유출을 방지하고, 나아가 정보 손실 매트릭스를 제공하여 사용자가 보다 적합한 익명화 수준을 선택할 수 있게 한다. 또한, 하둡 분산 환경 시스템을 이용하여 빅데이터 익명화 처리 속도를 높이기 위한 방법을 제시한다. 기존 데이터 배포 시스템과 본 논문에서 제안하는 시스템과의 차이점은 다음과 같다. 기존 시스템의 경우 사용자가 공유하고자 하는 테이블을 선택한 후, 익명화를 수행함으로써, 테이블 단위의 데이터 공유를 지원한다[17]. 그러나 본 논문에서 제안하는 시스템은, 사용자가 SQL문을 이용하여 배포하고자 하는 레코드들의 집합을 선택한 후, 익명화를 적용함으로써, 레코드 단위의 데이터 공유가 가능하다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 본 논문의 배경지식에 대하여 설명한다. 3장에서는 본 논문에서 개발한 시스템 구조 및 사용 예시에 대하여 설명한 후, 4장에서 결론을 맺는다.

2. 배경지식

2.1 k-익명성(k-Anonymity)

데이터 속에는 수많은 정보들이 담겨있다. 이 데이터 속에 포함된 속성 중 특정한 개인을 식별할 수 있는 속성을 '식별자(identifier)'라고 한다. 식별자의 예로는 주민등록번호, 휴대폰 번호 등이 있으며, 식별자는 개인 정보 유출을 직접적으로 유발하는 속성이다. 그래서 데이터 공유에 있어 식별자들은 제거되어야 한다. 하지만 식별자 외에도 다른 데이터(혹은 배경지식)과의 결합을 통해 개인 정보 유출을 유발할 가능성이 있는 속성이 있다. 이러한 속성을 '준식별자(quasi-identifier)'라고 한다[4,5]. 예를 들어, Fig. 1의 Original Table을 이용하여, 특정인 A의 민

Original Table			
RID	Gender	Age	Disease
1	M	21	Pneumonia
2	F	27	Diabetes
3	M	26	Anemia
4	F	34	Pneumonia
5	F	30	Anemia

2-Anonymized Table			
RID	Gender	Age	Disease
1	* (0~1)	20~29	Pneumonia
2	* (0~1)	20~29	Diabetes
3	* (0~1)	20~29	Anemia
4	* (0~1)	30~39	Pneumonia
5	* (0~1)	30~39	Anemia

Fig. 1. Original table and 2-Anonymized table

감한 정보(즉, 병명)을 추론하고자 하는 공격자를 가정하자. 이 공격자가 “A는 27살의 여성(Female)이고, Original Table에 A에 관한 정보가 포함되어 있다.” 라는 배경지식이 있으면, 이 공격자는 Original Table을 통해 “A는 당뇨병(diabetes)를 앓고 있다.” 라는 개인의 민감한 정보를 알아낼 수 있다. 이는 Original Table에 ‘27살의 여성(Female)“에 해당하는 레코드가 하나만 존재하기 때문이다. 이러한 준식별자는 개인 정보 유출의 가능성을 가지고 있으므로, 데이터 공유에 있어 비식별화 되어야 한다. 이러한 준식별자를 비식별화하는 가장 대표적인 프라이버시 모델이 k-익명성이다.

k-익명성은 일정한 규칙을 통하여 준식별자에 해당하는 값들을 상위 값으로 변환한다. 이러한 데이터 변환을 통해, 준식별자에 대해 동일한 속성 값을 가지는 데이터 집합인 동질 클래스(equivalence class)를 형성한다. k-익명성은 이러한 동질 클래스들의 크기를 k 이상으로 요구함으로써 특정 레코드를 동질 클래스내의 다른 $(k-1)$개의 레코드들과 구별할 수 없게 한다[4,5]. 예를 들어, Fig. 1은 원본 테이블과 2-익명화 테이블을 나타내고, 준식별자는 ‘Gender’와 ‘Age’이다. 2-익명화 테이블에는 1번~3번 레코드들로 구성된 동질 클래스와 4번~5번 레코드들로 구성된 동질 클래스가 존재한다. 앞서 설명한 것과 같이 2-익명성 테이블은 동질 클래스의 크기가 2 이상이므로 특정 레코드가 같은 동질 클래스 내의 다른 레코드들과 구별되지 않는다. 즉, 1번 레코드는 해당 레코드가 포함된 동질 클래스내의 다른 레코드(RID=2,3)와 속성 ‘Gender’와 ‘Age’의 값이 동일하므로, 다른 레코드들과 구별이 되지 않는다. 이처럼 k-익명성은 k개 이상의 레코드들을 동질 클래스로 일반화함으로써 익명성을 보장하는 기법이다.

2.2 일반화(Generalization) 기법

2.1절에서 설명한 바와 같이, k-익명화 기법은 데이터 일반화(generalization)를 기반으로 하고 있다. 데이터 일반화란 데이터를 해당 데이터의 상위 개념으로 변환하여 데이터를 추상화하는 방법이다. 이러한 데이터 일반화 기법에는 전역 일반화 방법(global generalization)과 지역 일반화 방법(local generalization)이 있다[13,14].

전역 일반화 방법은 원본 데이터를 일반화 격자(generalization lattice)를 통해 일반화 하는 방법이다. 일반화 격자는 각 속성의 범주 트리(taxonomy tree)의 단계로 구성되어 있다. 범주 트리는 계층적 트리로서, 상위 단계일수록 해당 속성의 높은 일반화 수준을 나타낸다. 예를 들어, Fig. 2의 (a)는 속성 ‘Age’와 ‘Gender’의 범주 트리이다. Fig. 2의 (a)에서와 같이 각각의 범주 트리는 3단계와 2단계의 일반화 수준을 가지고 있으며, 낮은 단계일수록 속성 원래의 값과 가깝고 높은 단계일수록 속성의 일반화의 범위가 크다. 그리고 Fig. 2의 (b)는 Fig. 2의 속성들의 범주 트리 단계를 통하여 구성된 일반화 격자이다. 일반화 격자 또한 낮은 단계일수록 일반화 데이터가 원본 데이터와 가깝고, 높은 단계일수록 원본 데이터의 변형이 심해진다. 그러므로 일반화 격자의 낮은 단계일수록 데이터 활용도는 증가하고 익명성의 단계는 낮아진다. 반대로 일반화 격자의 높은 단계일수록 데이터 활용도는 감소하지만 익명성의 단계는 높아진다. 따라서 전역 일반화의 가장 중요한 점은 각 속성의 적절한 분류 트리를 구성하고, 요구되는 익명화 수준에 따른 적합한 일반화 단계를 선택하는 것에 있다.

지역 일반화 방법은 전역 일반화 방법과 달리 일정한 규칙 없이 데이터의 유사성에 따라 데이터를

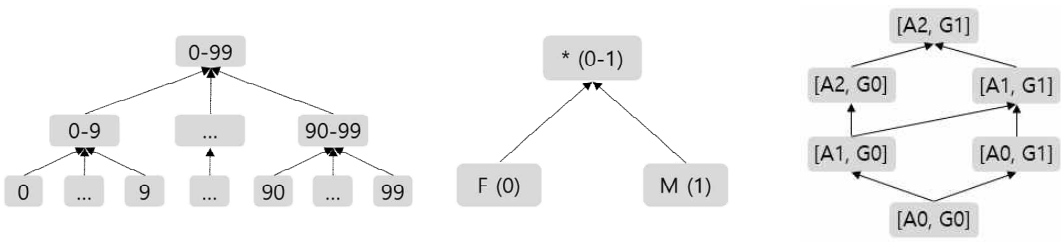

Fig. 2. Example of (a) Taxonomy Tree and (b) Generalization Lattice of ‘Age’ and ‘Gender’ Attributes.

일반화하는 방법이다. 이러한 지역 일반화 방법은 군집화를 통해 실시할 수 있으며, 이 때 각 군집이 하나의 동질 클래스로 형성된다[6]. 그렇기 때문에 지역 일반화 방법으로 형성된 동질 클래스는 하나의 동질 클래스에 포함되는 레코드의 수, 레코드의 일반화 범위 등이 각각 다를 수 있다. 그렇기 때문에 지역 일반화 방법은 사용자가 군집의 특성을 어떻게 설정하느냐에 따라 다양한 익명화 수준을 가질 수 있다.

2.3 정보 손실(Information Loss, IL)

일반화를 거친 데이터 값은 원래의 데이터 값과 어느 정도의 차이를 나타낸다. 이러한 차이를 정보 손실(information loss)라 한다. 그러므로 정보 손실 값은 그 값이 클수록 원본 데이터와 일반화 데이터 간의 차이가 크다는 것을 의미한다. 나아가 해당 속성들의 일반화 범위가 넓고, 결국 데이터 활용도가 낮음을 의미한다. 이러한 정보 손실 값을 이용하여 일반화 기법을 통해 익명화된 데이터가 원본 데이터로부터 얼마나 많이 변형되었는지 알 수 있고, 나아가 익명화된 데이터의 활용도 또한 알 수 있다.

동질 클래스 e가 m개의 속성으로 구성되어 있다고 가정하자. 이때, 정보 손실 값은 다음과 같이 정의된다[6].

$$IL(e) = |e| \cdot \sum_{i=1,\ldots,m} \frac{(MAX_{N_i} - MIN_{N_i})}{|N_i|} \quad (1)$$

여기서, N_i는 동질 클래스의 i-번째 속성을 나타내며, MAX와 MIN 값은 해당 속성에 대해 e에 속하는 레코드가 가질 수 있는 가장 큰 값과 작은 값을 나타낸다. 또한, $/N_i/$는 해당 속성 도메인의 크기를 나타낸다.

일반화된 테이블(AT)에 존재하는 모든 동질 클래스의 집합을 E라 가정하자. 이때, 일반화된 테이블에 대한 총 정보 손실 값은 다음과 같은 수식을 통해 구할 수 있다[6].

$$TOTALIL(AT) = \sum_{e \in E} IL(e) \quad (2)$$

3. 하둡 분산 환경 기반 프라이버시 보호 빅 데이터 배포 시스템

3.1 시스템 구조

본 연구에서 개발한 시스템은 데이터 소유자(즉, 사용자)가 데이터 배포에 있어 익명화된 데이터를 배포할 수 있도록 지원한다. Fig. 3은 본 논문에서 제안하는 프라이버시 보호 빅데이터 배포 시스템 구조이다. Fig. 3과 같이 사용자는 희망하는 익명화 수준(즉, k 값)과 배포를 원하는 데이터를 익명화 서버(Fig. 3의 점선 사각형)에 전달하다. 이때, 배포를 원하는 데이터는 일반적인 SQL 질의를 통해 표현된다. 익명화 서버는 사용자로부터 전달받은 SQL 질의를 데이터베이스(DB)를 이용하여 수행 한 후, 결과 값을 받아온다. 마지막으로 익명화 서버는 사용자가 전달한 k 값을 이용하여 SQL 질의 수행 결과 값에 대해 익명화를 실시하고(본 논문에서 제안하는 시스템은 전역 일반화 기법을 기반으로 익명화를 실시함), 익명화된 결과 값을 사용자에게 전달한다.

일반적으로 빅데이터 분석을 위해 공유하고자 하는 데이터들은 대용량 데이터에 해당한다. 또한, 데이터 익명화는 수행 시간이 오래 걸리는 작업에 해당한다. 그러므로 본 연구에서는 익명화 단계의 성능을 높이기 위해 하둡 분산 환경 시스템을 사용한다. 즉,

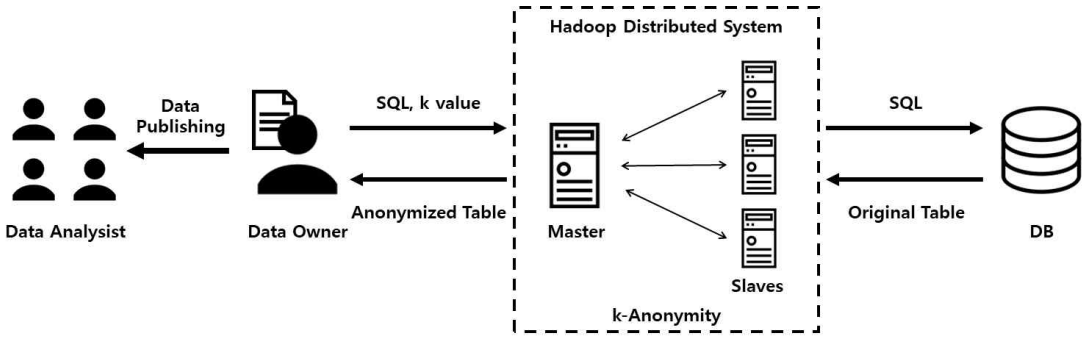

Fig. 3. A system architecture for the developed privacy-preserving big data publishing system.

익명화 서버에서 데이터 익명화는 대용량 처리에 적합한 Apache Hadoop[15]을 통해 실시한다. Apache Hadoop은 Map-Reduce를 통해 익명화를 진행한다. Fig. 4는 Apache Hadoop를 이용하여 k-익명화를 실시하는 과정을 나타낸 의사 코드에 해당되며, 크게 Main 단계, Map 단계, Reduce 단계로 구성되어 있다[18].

- Main 단계에서는 사용자로부터 요청받은 k 값과 분류 트리 그리고 원본 데이터를 입력받는다. 이 단계에서는 입력받은 분류 트리로 일반화 격자를 생성하고, 일반화 격자의 노드를 일반화 단계에 맞게 Map-Reduce에 전달한다.
- Map 단계에서는 원본 테이블과 일반화 격자의 노드를 정보를 읽는다. 그리고 각 레코드마다 준식별자와 민감 정보를 나눈 후, 준식별자를 일반화 격자 노드에 맞춰 동질 클래스로 일반화시킨다. 그리고 마지막으로 동질 클래스와 민감 정보로 변환된 레코드를 *key*와 *value*의 쌍에 해당하는 "(동질 클래스, 민감 정보)"의 형태로 결과 값을 Reduce에 전달한다.
- Reduce 단계에서는 Map 단계의 결과 값을 같은 *key* 값에 따라 *value* 값들을 취합한다. 즉, 같은 동질 클래스를 갖는 민감 정보들을 모아 하나의 리스트로 만든다. 이러한 Reduce 과정을 통해 "(동질 클래스, [민감 속성1, 민감 속성2, ...])"를 최종 결과 값으로 도출한다.
- 마지막으로, Main 단계에서는 Map-Reduce 과정을 통해 나온 결과 값의 각 동질 클래스에 대한 민감 속성의 개수가 k와 같거나 큰지 확인한다. 만약 각 동질 클래스에 대한 민감 속성들의 개수가 k보다 작을 경우, 이를 만족할 때까지 노드를 증가시키며 진행한다. 그리고 민감 속성의 개수가 k와 같거나 클 경우, k-익명성을 만족하므로 해당 결과를 Hadoop과 연동된 데이터베이스에 저장한다.

```
Main
Input: k-value, Taxonomy Tree, Original Table
Output: k-Anonymized Table
1.      Make_Generalization_Lattice(Taxonomy Trees);
2.      kcheck = false;
2.      while(!kcheck)
3.              node = next node in GL;
4.              output = Run_MapReduce(node);
5.              kcheck = Check_k_value(output); // (?>=k) - true
6.      end while;
7.      Enter_to_Database(output);
End.
```

```
Map class
Input: Original Table and Node of Generalization Lattice
Output: {Key, Value} = {Equivalent class, Sensitive value}
1.      line = a Recode of Original Table;
2.      while(hasNextLine)
3.              q = line's quasi-identifiers;
4.              s = line's sensitive value;
5.              eq = Generalization_by_Node(q, node);
6.              write(eq, s) ; // Output by {key, value}
7.      end while;
End.

Reduce class
Input: {Equivalent class, Sensitive value}
Output: {Equivalent class, Set of Sensitive values}
1.      while(hasNextValue)
2.              result += eq 's value;
3.      end while;
4.      write(eq, result);
End.
```

Fig. 4. The pseudocode of Map-Reduce for k-Anonymity process.

3.2 시스템 사용 예시

본 절에서는 본 연구에서 개발한 시스템의 활용성에 대하여 설명한다. 본 논문에서 제안하는 시스템은 Fig. 5와 같이 사용자(즉, 데이터 소유자)에게 두 가지 메뉴를 제공한다.

- Data Query: 첫 번째 메뉴는 사용자가 입력한 SQL 질의의 결과 레코드들(즉, 데이터 소유자가 배포하고자 하는 데이터베이스 내의 레코드들)에

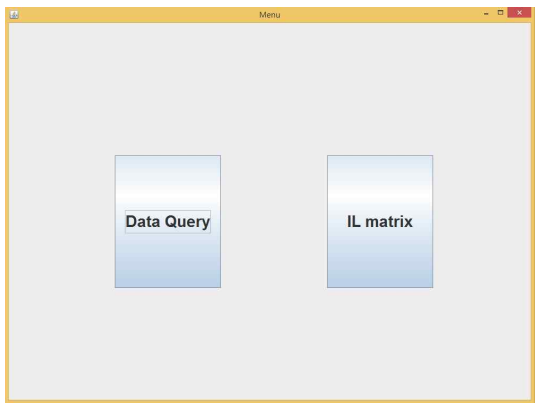

Fig. 5. The main menu of the developed privacy-preserving big data publishing system

대하여, 사용자가 지정한 k 값에 맞춰 익명화화는 기능이다 (3.2.1절).

- IL matrix : 두 번째 메뉴는 사용자가 적절한 익명화 단계의 k 값을 선택할 수 있도록 하는 기능으로서, 사용자가 지정한 일정 k 구간에 대하여 정보 손실 매트릭스를 제공해 주는 기능이다 (3.2.2절).

3.2.1 데이터 익명화 기능

사용자는 Fig. 6과 같이 배포를 희망하는 데이터를 SQL 질의를 통해 지정한다. 또한, 사용자는 배포하고자하는 데이터에 대한 익명화 수준(즉, k 값)을 입력할 수 있다. 해당 입력에 따른 결과를 익명화 서버에서 처리하여 Fig. 7와 같이 익명화 된 결과를 사용자에게 보여준다. 사용자는 해당 데이터를 Fig. 7의 우측 하단의 'save' 버튼을 통해 익명화 데이터를 저장한 후, 이를 일반에게 배포할 수 있다.

3.2.2 정보 손실 매트릭스 제공 기능

사용자는 배포를 원하는 데이터의 익명화 수준을 직접 선택해야한다. 하지만 k 값에 따라 익명화 수준은 달라지고, 그에 따라 데이터의 활용도 또한 변한다. 그렇기 때문에 빅데이터 공유에 있어서 배포를 희망하는 데이터에 대해 적합한 k 값을 선정하는 것이 매우 중요하다. 그래서 본 논문에서 제안하는 시스템은 사용자가 원하는 데이터를 사용자가 입력한 특정 범위의 k 값들에 따라 정보 손실을 측정하고, 이를 그래프로서 사용자에게 출력하여 사용자가 적합한 k 값을 선정할 수 있도록 지원한다.

예를 들어, Fig. 8은 사용자가 k=11~20 일 때, 사용자가 지정한 SQL 질의 결과에 대해 정보 손실 매트릭스를 요청하는 화면이다. 사용자는 SQL 질의문과 정보 손실 값을 확인하고자 하는 k 구간을 입력할 수 있다. 익명화 서버는 해당 요청에 따라 익명화 작업과 정보 손실을 측정하고 사용자에게 Fig. 9과 같이 보여준다. 사용자는 Fig. 9의 그래프를 보고 적합한 익명화 수준을 선택할 수 있다.

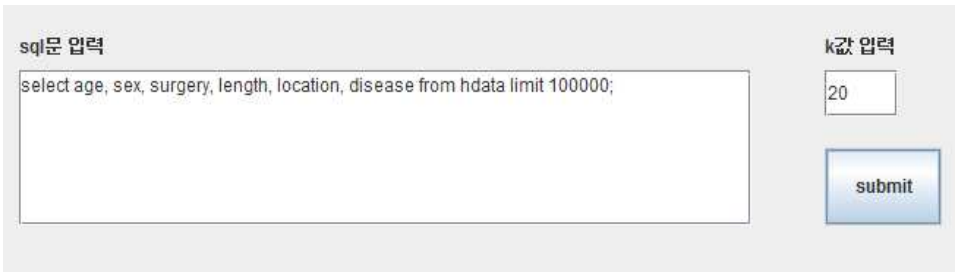

Fig. 6. An example of SQL query and anonymization level (k).

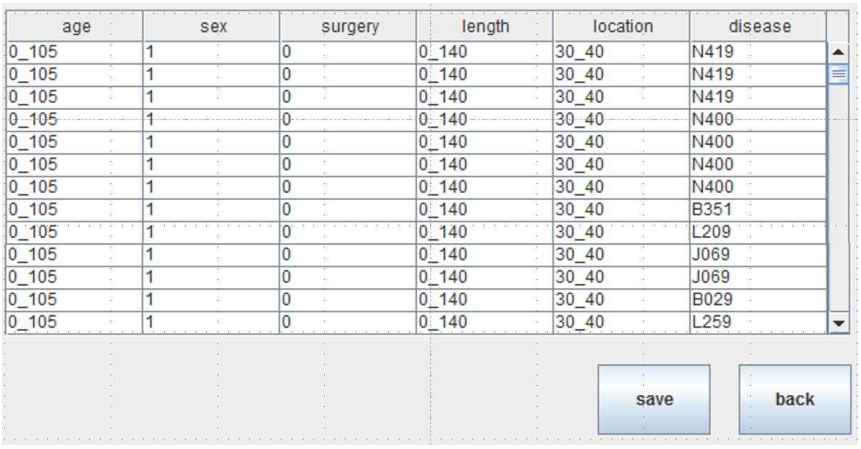

Fig. 7. The corresponding anonymized results of the SQL query and k value shown in Fig. 6.

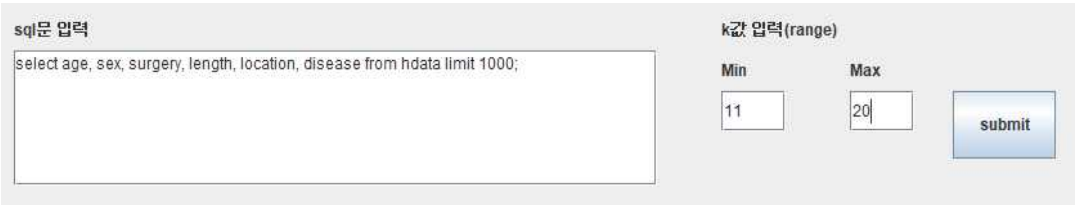

Fig. 8. An example of requesting information loss on varying *k*.

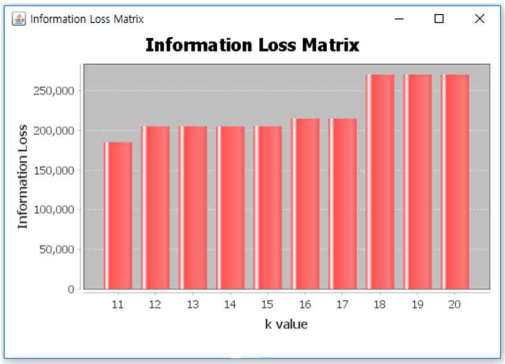

Fig. 9. Information Loss Matrix of the example request in Fig. 8.

4. 결론 및 향후 계획

본 논문에서는 데이터 소유자가 빅 데이터를 배포함에 있어 원하는 데이터를 *k*-익명화하여 제공함으로써 데이터 배포에서 발생할 수 있는 개인정보 유출을 방지하고, 나아가 정보 손실 매트릭스를 제공하여 사용자가 보다 적합한 익명화 수준을 선택할 수 있는 시스템을 개발했다. 본 논문에서 개발한 빅 데이터 배포 시스템을 활용함으로써, 데이터 소유자는 보다 쉽게 다양한 데이터를 배포할 수 있다. 또한, 그동안 데이터 소유자 직관에 의존했던 익명화 수준 선택과 달리, 본 논문에서 개발한 빅 데이터 배포 시스템은 정보 손실 요구 사항을 만족하는 최적의 익명화 수준 선택을 가능하게 함으로써, 배포된 데이터의 유용성을 최대화 할 수 있다.

향후 연구 계획은 현재 시스템에서 지원하고 있는 프라이버시 모델인 *k*-익명성 이외에, 다양한 프라이버시 모델(예, *l*-다양성, *t*-근접성)을 지원할 수 있도록 시스템을 확장하는 것이다. 또한 Apache Hadoop을 이용한 익명화 처리 과정을 메모리 기반 빅 데이터 처리 플랫폼인 Apache Spark[16]로 발전시켜, 익명화 처리 시간을 보다 단축시키고자 한다.

REFERENCE

[1] J. Kim, K. Jung, H. Lee, S. Kim, J. Kim, and Y. Chung, "Models for Privacy-preserving Data Publishing: A Survey," *Journal of Korean Institute of Information Scientists and Engineers,* Vol. 44, No. 2, pp. 195-207, 2017.

[2] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-preserving Data Publishing: A Survey of Recent Developments," *Association for Computing Machinery Computing Surveys,* Vol. 42, No. 4, pp. 14-53, 2010.

[3] N. Mohammed, B.C.M. Fung, P.C.K. Hung, and C.K. Lee, "Centralized and Distributed Anonymization for High-dimensional Healthcare Data," *Association for Computing Machinery Transactions on Knowledge Discovery from Data,* Vol. 4, No. 4, pp. 18-33, 2010.

[4] L. Sweeney, "K-anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* Vol. 10, Issue 05, pp. 557-570, 2002.

[5] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full Domain K-anonymity," *Proceedings of the Association for Computting Machinery Special Interest Group on Management of Data International Conference on Management of Data,* pp. 49-60, 2005.

[6] J. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient K-Anonymization Using Clustering Technique," *Proceeding of International*

Conference on Database Systems for Advanced Applications 2007: Advances in Databases: Concepts, Systems and Applications, pp. 188-200, 2007.

[7] K. Wang, P.S. Yu, and S. Chakraborty, "Bottom-up Generalization: A Data Mining Solution to Privacy Protection," *Proceedings of the IEEE International Conference on Data Mining*, pp. 249-256, 2004.

[8] B.C.M. Fung, K. Wang, and P.S. Yu, "Top-down Specialization for Information and Privacy Preservation," *Proceedings of the IEEE International Conference on Data Engineering,* pp. 205-216, 2005.

[9] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional K-anonymity," *Proceedings of the IEEE International Conference on Data Engineering,* pp. 25-35, 2006.

[10] G. Aggarwal, R. Panigrahy, T. Feder, D. Thomas, K. Kenthapadi, S. Khuller, et al., "Achieving Anonymity Via Clustering," *Association for Computing Machinery Transactions on Algorithms,* Vol. 6, No. 3 pp. 49-19, 2010.

[11] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy Beyond K-anonymity," *Association for Computing Machinery Transactions on Knowledge Discovery from Data,* Vol. 1, No. 1, pp. 3-52, 2007.

[12] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy Beyond K-anonymity and L-diversity," *Proceedings of the International Conference on Data Engineering,* pp. 106-115, 2007.

[13] S. Kim, H. Lee, and Y.D. Chung, "Privacy-preserving Data Cub for Electronic Medical Records: An Experimental Evaluation," *International Journal of Medical Informatics,* Vol 97, pp. 33-42, 2017.

[14] D.H. Kim and J.W. Kim, "A Study on Performing Join Queries over K-anonymous Tables," *Journal of The Korea Society of Computer and Information,* Vol. 22, No. 7, pp. 55-62, 2017.

[15] Apache Hadoop, http://hadoop.apache.org (accessed Sep., 1, 2017).

[16] Apache Spark, https://spark.apache.org (accessed Sep., 1, 2017).

[17] C. Dai, G. Ghinita, E. Bertino1, J.W. Byun, and N. Li, "TIAMAT: A Tool for Interactive Analysis of Microdata Anonymization Techniques," *Proceedings of the International Conference on Very Large Databases,* pp. 1618-1621, 2009.

[18] J.W. Kim, "Data Partitioning on MapReduce by Leveraging Data Utility," *Journal of Korea Multimedia Society,* Vol. 16, No. 5, pp. 657-666, 2013.

김 종 욱

1994년 3월~2000년 8월 고려 대학교 전산과학과 학사
2000년 9월~2002년 8월 한국과학기술원 전산학과 석사
2004년 1월~2009년 S12월 Arizona State University Computer Science 박사
2009년 10월~2010년 8월 Technicolor Member Research Staff
2010년 9월~2013년 8월 Teradata, Software Engineer
2013년 9월~현재 상명대학교 컴퓨터과학과 조교수

김 대 호

2011년 3월~2017년 2월 상명대학교 미디어소프트웨어학사
2017년 3월~현재 상명대학교 컴퓨터과학과 석사과정