情報保護學會論文誌
第18卷 第6(A)號, 2008. 12

프라이버시를 보호하는 동적 데이터의 재배포 기법*

이 주 창†, 안 성 준, 원 동 호, 김 응 모‡
성균관대학교 정보통신공학부

Privacy Preserving Data Publication of Dynamic Datasets*

Joochang Lee†, Sung Joon Ahn, Dong ho Won, Ung Mo Kim‡
School of Information and Communication Engineering Sungkyunkwan University

요 약

조직이나 기관에서 수집한 개인정보를 통계 분석, 공공 의료 연구 등을 목적으로 배포할 때는 데이터에 포함된 개인의 민감한 정보가 노출되지 않도록 보호해야 한다. 한편, 배포되는 데이터는 가능한 정확한 통계 정보를 제공해야 한다. *k*-anonymity와 *l*-diversity 모델은 이러한 프라이버시 침해 문제 해결을 위해 제안되었다. 그러나 두 모델은 데이터에 삽입과 삭제가 발생하지 않는 정적인 데이터를 단 한번 배포하는 상황을 가정하기 때문에 삽입과 삭제가 발생하는 동적인 데이터에 그대로 적용할 수 없다. 동적인 데이터의 프라이버시 보호 문제를 해결하기 위해 최근 m-invariance 모델이 제안되었다. 그러나 m-invariant 일반화 기법은 일반화로 인해 통계 정보로써 데이터의 품질을 저하시킨다는 단점이 있고, 배포된 데이터 중 일부 개인의 민감한 속성이 노출되었을 경우에 그 영향이 다른 부분으로 전이된다. 본 논문에서는 일반화를 사용하지 않으면서 간단한 삽입과 삭제 연산을 지원하는 동적 데이터의 배포 기법을 제안한다. 제안 기법은 데이터의 품질을 높이면서 m-invariance와 동등한 수준의 프라이버시 보호 정도를 제공한다.

ABSTRACT

The amount of personal information collected by organizations and government agencies is continuously increasing. When a data collector publishes personal information for research and other purposes, individuals' sensitive information should not be revealed. On the other hand, published data is also required to provide accurate statistical information for analysis. *k*-Anonymity and l-diversity models are popular approaches for privacy preserving data publication. However, they are limited to static data release. After a dataset is updated with insertions and deletions, a data collector cannot safely release up-to-date information. Recently, the m-invariance model has been proposed to support re-publication of dynamic datasets. However, the m-invariant generalization can cause high information loss. In addition, if the adversary already obtained sensitive values of some individuals before accessing released information, the m-invariance leads to severe privacy disclosure. In this paper, we propose a novel technique for safely releasing dynamic datasets. The proposed technique offers a simple and effective method for handling inserted and deleted records without generalization. It also gives equivalent degree of privacy preservation to the m-invariance model.

Keywords : data privacy, k-anonymity, m-invariance, generalization

Ⅰ. 서 론

정보통신기술의 발전으로 정보를 수집, 관리, 공유하기가 용이해짐에 따라 조직이나 기관에서는 정보시스템을 이용해 개인정보를 수집해 관리하고 있다. 수집된 개인정보를 연구나 통계 분석 등의 목적으로 배포할 때는

[표 1] 첫 번째 배포한 마이크로데이터

Name	Age	ZIP	Disease
철수	22	11000	간염
영호	28	12000	감기
민재	37	17000	폐렴
영희	30	21000	위궤양
지훈	33	23000	위염
수진	34	25000	당뇨
동원	39	26000	빈혈
은정	40	29000	골절
재영	46	31000	간염
유진	50	34000	폐암

(a) 마이크로데이터 T(1)

GID	Age	ZIP	Disease
1	[22,28]	[11k,12k]	간염
1	[22,28]	[11k,12k]	감기
2	[30,37]	[17k,21k]	폐렴
2	[30,37]	[17k,21k]	위궤양
3	[33,34]	[23k,25k]	위염
3	[33,34]	[23k,25k]	당뇨
4	[39,40]	[26k,29k]	빈혈
4	[39,40]	[26k,29k]	골절
5	[46,50]	[31k,34k]	간염
5	[46,50]	[31k,34k]	폐암

(b) 일반화 T'(1)

[표 2] 두 번째 배포한 마이크로데이터

Name	Age	ZIP	Disease
철수	22	11000	간염
영희	30	21000	위궤양
지훈	33	23000	위염
미연	29	22000	장염
동원	39	26000	빈혈
은정	40	29000	골절
재영	46	31000	간염
민석	43	47000	폐암
현석	51	38000	위암
정민	59	44000	당뇨

(a) 마이크로데이터 T(2)

GID	Age	ZIP	Disease
1	[22,30]	[11k,21k]	간염
1	[22,30]	[11k,21k]	위궤양
2	[29,33]	[22k,23k]	위염
2	[29,33]	[22k,23k]	장염
3	[39,40]	[26k,29k]	빈혈
3	[39,40]	[26k,29k]	골절
4	[43,46]	[31k,47k]	간염
4	[43,46]	[31k,47k]	폐암
5	[51,59]	[38k,44k]	위암
5	[51,29]	[38k,44k]	당뇨

(b) 일반화 T'(2)

데이터에 포함된 개인의 민감한 정보가 유출되지 않도록 보호해야 한다.

병원에서 [표 1a]와 같은 환자의 진료 기록을 공공 의료 연구를 위해 배포한다고 가정하자. [표 1a]처럼 미리 집계 요약되지 않은 테이블 형태의 데이터를 마이크로데이터(microdata)라 한다. 테이블에는 이름, 나이, ZIP, 병명 4개의 속성(attribute)이 있다. 이름은 개인을 유일하게 구별할 수 있는 식별자(identifier)이고 병명은 보호해야 할 민감한(sensitive) 속성이다. 데이터를 배포할 때 프라이버시 보호를 위해서 식별자를 테이블에서 제거하고 나이, ZIP, 병명으로만 구성된 테이블을 배포할 수 있지만 이런 방법만으로는 프라이버시를 충분히 보호할 수 없다[12].

만일 공격자가 철수의 나이, ZIP과 병원에서 진료를 받았다는 사실을 알고 있으면 공격자는 철수의 병명이 간염이라는 사실을 알 수 있다. 나이, ZIP과 같이 외부 정보와 연결되어 개인을 식별하는데 이용될 수 있는 속성을 quasi-identifier(QI)라 한다.

이러한 프라이버시 침해 문제를 해결하기 위해 *k*-anonymity[12] 모델과 *l*-diversity[8] 모델이 제안되었다. *k*-anonymity 모델은 테이블에서 각 레코드가 최소 *k*-1개의 서로 다른 레코드들과 구별되지 않도록 하여 익명성을 제공한다. *k*-anonymity 프라이버시 요구사항은 quasi-identifier 값을 덜 구체적인 값으로 일반화(generalization) 하거나 제거(suppression)하여 레코드 집합을 같은 quasi-identifier 값을 가지는 그룹으로 나누는 기법을 사용함으로써 만족시킬 수 있다[13]. 테이블에서 같은 quasi-identifier를 가지는 레코드들의 집합을 동등클래스(equivalent class)라 한다. *k*-anonymity는 익명성을 제공하기는 하지만 동등클래스 내에서 민감한 속성의 분포는 고려하지 않는다는 문제점이 있다. 동등클래스 안에서 민감한 속성의 값이 모두 같다면 공격자는 개인의 민감한 속성을 정확하게 알아낼 수 있다. *l*-diversity는 이러한 문제점을 해결하기 위해 동등클래스 안에서 가장 빈번하게 발생하는 민감한 속성의 확률이 최대 1/l이 되도록 하여 이러한 문제점을 해결한다. [표 1b]는 2-anonymity와 2-diversity를 만족하는 테이블이다. k나 l 값이 커질수록 프라이버시 보호 정도도 증가한다. 그러나 일반화로 인한 정보손실로 데이터의 유용성은 감소한다. 데이터의 유용성은 익명화된 데이터를 대상으로 COUNT, SUM과 같은 SQL 집합 쿼리(aggregate query)를 수행한 결과가 원본 데이터의 결과와 가까울수록 높다.

종래의 프라이버시 보호 기법은 정적인(static) 데이터를 배포하는 상황, 즉, 데이터를 배포하는 순간에 모든 데이터가 준비되어 단 한번 배포하는 상황을 가정하고 있다. 그러나 지속적으로 데이터가 변경되는 환경에서 최근의 정보를 제공하기 위해서는 데이터가 삽입되거나 삭제되는 경우를 고려해야 한다. 예를 들어, 병원에서 환자 진료 기록을 매월 또는 분기별로 배포할 수 있는데, 이럴 경우 새로 입원한 환자와 퇴원한 환자로 인해 진료 기록에 삽입과 삭제가 발생한다. 데이터의 추

접수일 : 2008년 3월 26일; 수정일 : 2008년 10월 18일;
채택일 : 2008년 11월 18일
* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(IITA-2008-C1090-0801-0028) 및 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천 기반 기술 개발 사업의 연구결과로 수행되었음.
† 주저자, lordeath@ece.skku.ac.kr
‡ 교신저자, umkim@ece.skku.ac.kr

세나 경향을 분석하기 위해서는 삽입/삭제가 발생하는 동적데이터를 안전하게 배포할 수 있는 기법이 필요하다. 이러한 동적(dynamic) 데이터에 대해 기존 기법을 그대로 적용할 경우 이전에 배포된 데이터와 새로 배포된 데이터 간의 연관관계로 인해 프라이버시가 침해될 수 있다.

예제 1 [표 1a]과 같은 마이크로데이터를 [표 1b]와 같이 익명화하여 처음으로 배포하였다고 가정하자. 이후 데이터에 삽입/삭제가 발생하여 두 번째로 배포할 시점의 마이크로데이터가 [표 2a]와 같이 변경되었고 이를 익명화 하여 [표 2b]와 같은 데이블을 배포하였다. [표 1a]에서 삭제된 레코드와 [표 2a]에 추가된 레코드들을 기울임꼴로 표시하였다. 두 테이블은 모두 독립적으로 2-anonymous하고 2-diverse 하지만 공격자는 두 테이블 간의 연관관계를 이용해 다음과 같이 추론할 수 있다. 공격자는 철수의 quasi-identifier 값과 철수의 레코드가 두 테이블에 모두 포함되어 있다는 사실을 배경지식으로 알고 있다. 공격자는 첫 번째 배포된 [표 1b]로부터 철수의 병명이 간염 또는 감기라는 사실을 알 수 있다. 또한 [표 2b]로부터 간염 또는 폐렴이라는 사실을 알 수 있다. 결국 두 사실을 결합하여 철수의 병명이 간염이라고 결정한다. [표 2b]를 다른 방법으로 일반화하더라도 철수의 병명이 간염이라는 사실은 항상 노출된다. [표 1b]에서 철수가 가질 수 있는 병명은 간염과 감기 두 가지 경우였는데 데이터에 변경이 일어나 [표 2b]에는 감기를 병명으로 가지는 레코드가 없다. 이러한 현상을 치명적 결여(critical absence)라 한다[16]. 치명적 결여는 레코드가 삭제되는 경우에만 발생한다. 삽입만 발생하는 경우에는 이러한 현상이 일어나지 않는다.

m-invariance 모델은 삽입/삭제가 발생하는 동적인 데이터의 재배포를 지원한다[16]. [표 1]과 [표 2]에서 재영의 레코드는 민감한 속성이 노출될 확률은 동일하다. 두 번째로 배포된 테이블 [표 2b]에는 재영과 같은 동등 클래스에 속했던 유진의 레코드가 삭제되고 민석의 레코드가 삽입되었지만 재영이 가질 수 있는 병명은 간염 또는 폐암으로 유지되었다. 즉, 여러번 배포된 테이블에서 어떤 레코드가 가질 수 있는 민감한 속성의 값들이 변하지 않으면 데이터를 재배포 하더라도 민감한 속성 값이 노출될 확률에는 변화가 없다. m-invariant 일반화 기법은 삭제되는 레코드와 삽입되는 레코드의 민감한 속성이 같으면 삭제되는 레코드를 삽입된 레코드로 대체하여 다시 일반화하고 치명적 결여가 발생했을 때는 모조레코드(counterfeit)를 삽입한다. m-invariant 일반화 기법으로 두 번째 마이크로데이터를 익명화한 테이블은 [표 3]과 같다. m-invariant 일반화 기법은 익명화한 테이블 [표 3a]와 함께 동등 클래스에 삽입된 모조 레코드의 개수를 저장하고 있는 보조 테이블 [표 3b]를 같이 배포하여 데이터 분석의 정확성을 높인다.

[표 3] T⑵의 m-invariant 일반화

Name	GID	Age	ZIP	Disease
철수	1	[22,30]	[11k,21k]	간염
c1	1	[22,30]	[11k,21k]	감기
c2	2	[30,37]	[17k,21k]	폐렴
영희	2	[30,37]	[17k,21k]	위궤양
지훈	3	[33,59]	[23k,44k]	위염
정민	3	[33,59]	[23k,44k]	당뇨
동원	4	[39,40]	[26k,29k]	빈혈
은정	4	[39,40]	[26k,29k]	골절
재영	5	[43,46]	[31k,47k]	간염
민석	5	[43,46]	[31k,47k]	폐암
미연	6	[29,51]	[22k,38k]	장염
현석	6	[29,51]	[22k,38k]	위암

(a) 모조레코드를 이용한 일반화

GID	Count
1	1
2	1

(b) 모조레코드 통계

1.1 m-invariance의 문제점

m-invariance 기법의 문제점은 다음과 같다. 첫째, 일반화로 인한 정보의 손실이 발생한다. quasi-identifier를 일반화하기 때문에 통계 정보로써의 가치가 떨어진다. 둘째, 어떤 레코드의 민감한 속성이 공격자에게 노출된 경우에 같은 동등클래스에 속한 다른 레코드들에 영향을 미친다. 또한 그 영향은 이전에 배포되었던 테이블로 전파된다. [표 3a]에서 지훈의 병명이 위염이라는 사실을 공격자가 이미 알고 있다고 가정한다. 이 사실로부터 공격자는 지훈과 같은 동등클래스에 속한 정민의 병명이 당뇨라는 사실을 알 수 있다. 뿐만 아니라 이전에 배포되었던 테이블인 [표 1b]에서 지훈과 같은 동등클래스에 속한 수진의 병명이 당뇨라는 사실 또한 알 수 있다.

1.2 제안 기법의 필요성

일반화에 기반한 익명화 기법은 전체 레코드 집합을 여러 개의 같은 quasi-identifier를 가지는 동등클래스로

[표 4] QIT-PT 익명화 QIT(1), PT(1)

Name	Age	ZIP	Disease
철수	22	11000	간염(0.5), 골절(0.5)
영호	28	12000	감기(0.5), 위궤양(0.5)
민재	37	17000	빈혈(0.5), 폐렴(0.5)
영희	30	21000	간염(0.5), 위궤양(0.5)
지훈	33	23000	감기(0.5), 위염(0.5)
수진	34	25000	간염(0.5), 당뇨(0.5)
동원	39	26000	빈혈(0.5), 폐암(0.5)
은정	40	29000	골절(0.5), 폐렴(0.5)
재영	46	31000	간염(0.5), 위염(0.5)
유진	50	34000	당뇨(0.5), 폐암(0.5)

[표 5] QIT-PT 익명화 QIT(2), PT(2)

Name	Age	ZIP	Disease
철수	22	11000	간염(0.5), 골절(0.5)
영희	30	21000	간염(0.5), 위궤양(0.5)
지훈	33	23000	감기(0.5), 위염(0.5)
동원	39	26000	빈혈(0.5), 폐암(0.5)
은정	40	29000	골절(0.5), 폐렴(0.5)
재영	46	31000	간염(0.5), 위염(0.5)
민석	43	47000	당뇨(0.5), 폐암(0.5)
정민	59	44000	간염(0.5), 당뇨(0.5)
미연	29	22000	위암(0.5), 장염(0.5)
현석	51	38000	위암(0.5), 장염(0.5)

분할한다. 일반화 기법의 문제점은 레코드의 익명성이 같은 동등클래스에 속한 다른 레코드에 의존한다는 것이다. 어떤 동등클래스에 삽입/삭제 연산이 발생하면 같은 동등클래스에 속한 다른 레코드에 영향을 미친다. 제안 기법은 레코드 집합을 분할하는 대신 각 레코드마다 가질 수 있는 민감한 속성의 확률을 따로 저장한다. [표 1a]와 [표 2a]를 각각 익명화한 결과는 각각 [표 4]와 [표 5]와 같다. [표 4]와 [표 5]에서 민감한 속성인 병명이 가질 수 있는 값과 확률은 실질적으로 [표 6]과 같은 형태로 저장된다. 제안 기법은 원본 테이블을 quasi-identifier를 저장하는 QIT(Quasi-Identifier Table)와 PT(Probability Table)로 분리하여 레코드마다 가질 수 있는 민감한 속성 값들을 확률과 함께 저장한다. 두 테이블에는 Row_ID 컬럼이 추가되어 조인(join) 연산에 사용된다.

제안 기법은 부식별자를 일반화 하지 않기 때문에 일반화에 기반한 익명화 기법에 비해 데이터 분석에 더 유리하다. [표 3]과 [표 5]를 대상으로 Q1과 같은 SQL 쿼리를 실행한 결과는 다음과 같다..

[표 6] QIT-PT 익명화

Name	Age	ZIP	Row_ID
철수	22	11000	1
영호	28	12000	2
민재	37	17000	3
영희	30	21000	4
지훈	33	23000	5
수진	34	25000	6
동원	39	26000	7
은정	40	29000	8
재영	46	31000	9
유진	50	34000	10

(a) Quasi-identifier 테이블(QIT)

Row_ID	Disease	Prob
1	간염	0.5
1	골절	0.5
2	감기	0.5
2	위궤양	0.5
3	빈혈	0.5
3	폐렴	0.5
4	간염	0.5
4	위궤양	0.5
...	...	...
10	당뇨	0.5
10	폐암	0.5

(b) 확률 테이블(PT)

Q1: SELECT COUNT(*) FROM Microdata
WHERE Age <= 30 AND Zipcode IN
[15001, 20000] AND Disease = '감기'

[표 2a]에서 볼 수 있듯이 원본 테이블에서 쿼리의 결과는 0이다. [표 3]에서는 첫 번째 동등클래스의 레코드들이 Age 조건을 만족하고 Zipcode 조건을 부분적으로 만족한다. Zipcode가 일반화 된 구간 [11k, 21k]에서 값들의 분포가 균등하다고 가정하면 레코드가 [15k, 20k] 구간에 있을 확률은 $\frac{20k-15k}{21k-11k}=\frac{1}{2}$이라 유추할 수 있고 동등클래스 내에서 간염이 한번 발생하므로 예측한 쿼리 결과는 0.5이다. 반면에 [표 5]는 quasi-identifier의 정확한 값을 알 수 있기 때문에 원본 테이블의 쿼리 결과와 같은 0을 얻을 수 있다.

공격자가 사전에 특정 개인의 민감한 속성 값을 이미 알고 있을 경우에도 제안 기법은 안전하다. 1.1절에서 언급한 것과 동일한 상황이 발생하더라도 제안 기법에서는 레코드마다 민감한 속성 값을 따로 저장하여 의존성을 제거하였기 때문에 프라이버시 침해의 영향이 다른 레코드나 이전에 배포되었던 테이블로 전파되지 않는다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구 동향을 소개한다. 3절에서는 동적 데이터의 안전한 재배포 문제를 정의하고 익명화 알고리즘을 제안하며 안전성과 데이터 유용성에 대해 분석한다. 마지막으로 4절에서는 결론과 향후 연구 과제를 기술한다.

Ⅱ. 관련 연구

k-anonymity 모델과 일반화를 이용한 프라이버시를 보호 기법은 Sweeney와 Samarati에 의해 소개되었다

[11, 12]. 이후 *k*-anonymity 모델을 바탕으로 여러 익명화 알고리즘이 제안되었다[13, 11, 5, 2, 6, 4, 10]. 일반화로 인한 정보 손실과 데이터의 유용성은 서로 상충하는 관계에 있으며 요구되는 익명성을 제공하면서 최소한의 정보 손실을 가지도록 최적의(optimal) 일반화 결과를 구하는 문제는 NP-hard 문제임이 증명되었다[9].

k-anonymity 모델은 익명성을 제공하지만 동등클래스 내의 민감한 속성 값의 분포는 고려하지 않는 문제가 있다. *l*-diversity 모델은 이러한 문제점을 지적하고 동등클래스 안의 레코드들이 가질 수 있는 민감한 속성 값이 *l*개의 잘 표현된(well-represented) 값을 가지도록 요구하고 distinct, entropy, recursive *l*-diversity를 제안하였다[8]. 대표적인 *l*-diversity 모델은 동등클래스 내에서 가장 빈번히 발생하는 민감한 속성 값이 최대 1/*l* 이하가 되도록 요구한다. *l*-diversity 모델이 동등클래스 내의 민감한 속성 분포를 고려하기는 하지만 민감한 속성 값의 분포가 한쪽으로 치우치거나 의미(semantic)가 유사한 경우에는 충분히 프라이버시를 보호하지 못한다. Li et al.은 이러한 문제를 해결하기 위해 동등클래스 내의 민감한 속성 값의 확률 분포를 전체 데이터의 확률 분포와 근접하도록 할 것을 제안하였다[7].

일반화에 기반한 익명화 기법은 quasi-identifier를 일반화함으로써 발생하는 정보 손실이 단점이다. 최근 일반화 대신 레코드들을 그룹핑하여 같은 그룹 내에서 민감한 속성 값을 퍼뮤테이션(permutation) 하는 기법[18]과 원본 테이블을 quasi-identifier와 민감한 속성을 저장하는 2개의 테이블로 분리하는 기법[15]이 제안되었다. 이러한 기법은 개인이 데이터에 포함되어 있다는 사실을 감출 수는 없지만 공격자가 개인이 이미 배포된 데이터에 포함되어 있다는 사실을 배경지식으로 가지고 있으면 안전성은 일반화 기법과 동등하다[15]. 이러한 익명화 기법들은 quasi-identifier를 일반화하지 않기 때문에 데이터 분석에 보다 높은 정확성을 제공하는 것이 장점이다.

기존의 익명화 기법의 대부분은 데이터를 단 한번 배포하는 것을 가정하고 있다. Byun et al.은 최초로 삽입이 발생하는 점진적인 데이터에서 익명성을 유지할 수 있는 기법을 제안하였다[1]. Fung et al.도 삽입 연산이 발생하는 데이터의 익명화 방법을 제안 하였으나 역시 삭제 연산을 고려하지 않는다[3]. Wang et al.은 같은 테이블을 다른 형태의 뷰(view)로 여러 번 배포할 때 프라이버시 보호 기법을 제안하였지만 삽입/삭제 연산은 고려하지 않는다[14]. 본 논문에서 다루고 있는 삽입/삭제 연산이 발생하는 동적 데이터의 프라이버시 보호 문제는 m-invariance에서 다루고 있다[16]. 그러나 m-invariance는 일반화에 기반하기 때문에 이로 인한 정보 손실이 발생하고 특정 개인의 프라이버시가 침해되었을 때 그 영향이 다른 개인의 프라이버시 침해로 전이 될 수 있다는 문제점이 있다. 국내 연구로는 삽입/삭제 연산을 지원하면서 *l*-diversity를 유지하는 기법이 있다[19]. 그러나 *l*-diversity를 유지하는 것만으로는 시그니처 변경으로 인해 민감한 속성 값이 노출되는 것을 방지할 수 없다. 제안 기법은 일반화를 사용하지 않고 퍼뮤테이션에 기반한 익명화 기법처럼 quasi-identifier를 그대로 저장하고 개인이 가질 수 있는 민감한 속성 값을 확률과 함께 별도의 테이블에 저장함으로써 데이터의 유용성을 높이면서 삽입/삭제 연산이 발생하는 동적 데이터를 효과적으로 익명화 할 수 있는 방법이다.

Ⅲ. 제안 기법

본 절에서는 삽입/삭제가 발생하는 동적 데이터를 프라이버시 침해 없이 익명화 하는 알고리즘을 제안하고 제안하는 알고리즘이 어떻게 프라이버시를 보호하는지 기술한다.

3.1 문제 정의 및 알고리즘

개인정보를 담고 있는 마이크로데이터를 $T=\{t_1,...,t_n\}$라 하자. T의 카디널리티(cardinality)를 $|T|=n$으로 표기한다. T의 컬럼은 m개의 quasi-identifier 속성 $QI_1,...,QI_m$과 한 개의 민감한 속성 S로 구성된다. 민감한 속성 S는 이산적인(discrete) 값을 가지는 범주형(categorical) 데이터라 가정한다. 민감한 속성 값은 S의 도메인 $D=\{s_1,...,s_l\}$에서 한 개의 값을 가진다. 도메인 D의 크기를 $|D|=l$이라 표기한다. quasi-identifier $QI_1,...,QI_m$는 범주형 또는 수치형(numeric) 데이터가 될 수 있다. 레코드 $t\in T$ 의 속성 QI_i의 값을 $t[QI_i]$, $(0\le i\le m)$, 민감한 속성 값을 $t[S]$라 표기한다. 시간이 지남에 따라 T에 삽입/삭제 연산이 발생한다. 데이터 소유자는 특정 시간에 프라이버시 침해 없이 T를 익명화하여 배포하고자 한다. j번째 배포할 시점에 T의 스

냅샷(snapshot)을 $T(j)$로 표기한다.

정의 1 (Quasi-identifier) Quasi-identifier는 T의 컬럼 $QI_1,...,QI_m$의 집합으로 외부 정보와 연결되어 테이블에 포함된 개인을 식별함으로써 민감한 속성 S의 값을 추론하기 위해 사용된다.

정의 2 (일반화) 일반화 기법은 T의 레코드 집합을 파티션 하여 같은 l개의 동등클래스를 생성한다. 레코드 $t \in T$은 일반화된 테이블 T^*에서 다음과 같은 형태의 레코드를 가진다.

$$(E_k[QI_1], E_k[QI_2], ..., E_k[QI_m], t[S])$$

E_k $(1 \le k \le l)$는 일반화된 데이블에서 t를 포함하는 유일한 동등클래스를 의미한다. $E_k[QI_i]$ $(1 \le i \le m)$는 $t[QI_i]$을 덜 구체적인 값으로 일반화한 값이다. 같은 동등클래스에 속하는 모든 $t \in E_k$에 대해 $E_k[QI_i]$ 값은 동일하다.

정의 3 (QIT-PT 익명화) QIT-PT 익명화 기법은 T를 익명화하여 두 테이블 QIT와 PT를 생성한다. QIT의 스키마는 다음과 같다.

$$QIT = (QI_1, .QI_m, Row\_ID)$$

모든 $t_i \in T$, $(0 \le i \le n)$에 대해 QIT는 $(t_i[QI_1], t_i[QI_m], Row\_ID)$ 투플을 가진다. PT의 스키마는 다음과 같다.

$$PT = (Row\_ID, S, PROB)$$

QIT에는 quasi-identifier가 저장되고 PT에는 보호해야할 민감한 속성이 확률과 함께 저장된다. Row_ID 컬럼은 조인(join) 연산에 사용된다. PT 테이블에서 같은 Row_ID를 가지는 레코드들의 $PROB$ 컬럼의 합은 1.0 이다.

m-invariance 모델은 변경이 발생하는 마이크로데이터를 재배포할 때 프라이버시 보호를 위해 제안 되었으며 시그니쳐(signature)와 m-uniqueness 개념에 기반한다[16].

정의 4 (시그니처) 마이크로데이터 T의 파티션을 P라 하고 t를 동등클래스 $E \in P$에 속하는 레코드라 하자. P에서 t의 시그니처는 E에서 중복되지 않는 민감한 속성 값의 집합이다.

정의 5 (m-uniqueness) 일반화를 사용하여 익명화된 테이블에서 어떤 동등클래스 내의 민감한 속성 값이 최소 m개 이상이고 그 값이 모두 다르면 그 동등클래스는 m-unique 하다. 테이블의 모든 동등클래스가 m-unique 하면 그 테이블은 m-unique 하다.

정의 6 (m-invariance 프라이버시 요구사항) 삽입/삭제가 일어나는 동적데이터를 일반화하여 배포된 테이블 $T^*(1),...,T^*(n)$가 모두 m-unique하고 $t \in T(i)$ $(1 \le i \le n)$가 포함된 모든 $T^*(x),...T^*(y)$ $(1 \le x < y \le n)$에서 t의 시그니처가 동일하면 그 일반화 기법은 m-invariant 하다.

정의 7 (공격자의 배경 지식) 공격자는 공격의 대상이 되는 개인의 quasi-identifier 값을 알고 있으며 이에 해당되는 레코드가 어떤 $QIT(j)$와 $PT(j)$에 포함되어 있는지를 배경 지식으로 알고 있다. 즉, 몇 번째 배포한 마이크로데이터에 포함되어 있는지를 알고 있다. 공격자의 배경 지식을 B라 한다.

일반화 기법이 m-invariance 원리를 만족하면 공격자가 B를 이용해 배포된 마이크로데이터에 포함된 개인의 민감한 속성을 유추할 수 있는 확률은 최대 $1/m$ 이다[16].

정의 8 (프라이버시 침해) 공격자가 배경지식 B와 배포된 마이크로데이터 $\{QIT(1), PT(1)\}$, ..., $\{QIT(n), PT(n)\}$를 이용해 개인의 민감한 속성 S의 값을 $1/m$ 보다 큰 확률로 유추할 수 있으면 프라이버시 침해가 발생한다고 정의하며 이는 다음과 같은 조건부 확률로 나타낼 수 있다.

$$P(t[S] = v \mid \bigcup_{i=1}^{n} \{QIT(i), PT(i)\}, B) > 1/m$$

정의 9 (프라이버시를 보호하는 동적 데이터의 재배포 문제) $n-1$개의 먼저 배포된 마이크로데이터 $\{QIT(1),PT(1)\},...,\{QIT(n-1),PT(n-1)\}$이 있을 때 프라이버시 침해가 발생하지 않도록 $T(n)$을 $\{QIT(n),PT(n)\}$으로 익명화 하는 것이다.

데이터 재배포로 인해 프라이버시 침해가 발생하는 이유는 데이터에 변경으로 인해 배포된 테이블에 포함된 개인의 레코드가 가질 수 있는 민감한 속성의 값이나 확률이 달라지기 때문이다. m-invariant 일반화는 배포된 마이크로데이터 버전들이 m-unique하고 시그니처가 동일하도록 함으로써 재배포로 인해 민감한 속성 값이 노출되는 것을 방지한다. 어떤 개인의 레코드가 x번째부터 y번째까지 $(1 \le x < y \le n)$ 배포된 마이크로데이터에 포함되어 있다고 하자. 데이터 재배포로 인한 프라이버시 침해를 방지하기 위해선 $t \in QIT(i)$, $(x \le i \le y)$가 가질 수 있는 민감한 속성의 종류와 확률이 모든 $QIT(i)$에서 동일해야만 한다.

제안 기법은 민감한 속성 값이 노출되는 것을 방지하기 위해 종래 일반화 기법처럼 테이블을 파티션하는 대신 민감한 속성 값에 노이즈를 더한다. S의 도메인 D에서 균등한(uniform) 확률로 임의의 값을 $m-1$개 발생시켜 민감한 속성 값에 추가한다. 이렇게 하면 공격자는 어떤 값이 원래 값인지 구별할 수 없다. 데이터를 재배포 하는 경우 기존에 배포된 마이크로데이터에 포함된 레코드는 새로 배포될 테이블에 그대로 복사하여 시그니처가 변경되지 않도록 한다. 새로 삽입된 레코드는 마찬가지로 임의의 노이즈를 더해 테이블에 추가한다. 즉, 처음 삽입되는 레코드들은 노이즈를 더하고 이들 레코드가 다음에 배포될 데이터에도 포함되면 시그니처를 동일하게 유지시킨다.

익명화 알고리즘의 의사 코드는 [그림 1]과 같다. 알고리즘은 $T(n-1)$, $T(n)$, $QIT(n-1)$, $PT(n-1)$, 그리고 m-invariance 모델의 파라미터 m을 입력받아 $QIT(n)$, $PT(n)$를 구한다. $T(n-1) \cap T(n)$에 해당하는 레코드들은 $QIT(n-1)$, $PT(n-1)$에서 $QIT(n)$, $PT(n)$로 그대로 복사된다. 삭제되는 레코드 $T(n-1) - T(n)$는 모두 삭제된다. 새로 삽입된 레코드들 $T(n) - T(n-1)$은 민감한 속성 값에 노이즈를 더해 $QIT(n)$, $PT(n)$에 입력한다. [그림 1]에 제시된 알고리즘은 삭제된 레코드 제거, 노이즈 추가, 삽입된 레코드 추가하는 부분이 분리되어 있으나 세 작업은 데이터를 한번 스캔하면서 동시에 수행될 수 있다. 따라서 제안 알고리즘의 시간 복잡도는 $O(n)$이며 기존 일반화 기법들이 정보손실을 최소화하기 위해 여러 번의 스캔을 필요로 하는 것과 비교해 효율적이다.

3.2 분석

3.2.1 안전성

4.1절에서 기술한 바와 같이 QIT-PT 익명화 기법은 민감한 속성 값에 노이즈를 더해 레코드가 가질 수 있는 민감한 속성의 종류가 m개가 되도록 하여 공격자가 quasi-identifier를 이용해 민감한 속성을 정확히 알 수 없도록 한다. 또한 데이터 재배포로 인한 프라이버시 침해를 방지하기 위해 배포된 마이크로데이터에서 레코드의 시그니처를 동일하게 유지한다. 따라서 QIT-PT 익명화 기법은 프라이버시 침해 없이 효과적으로 삽입/삭제가 일어나는 데이터를 재배포할 수 있는 기법이다.

정리 1 QIT-PT 익명화 기법은 m-invariance 프라이버시 요구사항을 만족한다.

증명 일반화에 적용된 m-invariance 프라이버시 요구사항은 일반화되어 배포된 테이블들이 모두 m-uniqueness를 만족하고 같은 레코드가 여러 버전에 포함되어 있을 때 시그니처가 항상 동일해야 한다. QIT-PT 익명화 기법은 m-invariance의 파라미터 m을 입력받아 원본 마이크로데이터의 민감한 속성 S 값에 임의로 발생시킨 노이즈를 더해 각 레코드가 가질 수 있는 민감한 속성 값이 m개가 되도록 한다. 따라서 QIT-PT 익명화 기법은 m-uniqueness를 만족한다. 또한 시그니처를 동일하게 유지하기 위해서 이전에 배포된 테이블에 포함되었던 레코드가 삭제되지 않았으면 이번에 배포할 테이블에 그대로 복사한다. 따라서 QIT-PT 익명화 기법의 프라이버시 보호 정도는 m-invariant 일반화와 동등하다.

QIT-PT 익명화 기법은 m-invariance 프라이버시 요구사항을 만족하기 때문에 공격자가 정의 7의 배경지식 B를 이용해 데이터에 포함된 개인의 민감한 속성 값을 알아낼 확률은 최대 $1/m$이다.

```
Input:  T(n−1): (n-1)th snapshot
        T(n): nth snapshot
        QIT(n−1): quasi-identifier table for (n-1)th release
        PT(n−1): probability table for (n-1)th release
        m: parameter for m-invariance privacy requirement

Output: QIT(n): quasi-identifier table for nth release
        PT(n): probability table for nth release

// remove deleted tuples
1.  SM = T(n−1) − T(n)
2.  for each t∈SM do
3.      delete t from QIT(n−1) and PT(n−1)

4.  copy all tuples of QIT(n−1) to QIT(n)
5.  copy all tuples of PT(n−1) to PT(n)

6.  rid = SELECT MAX(Row_ID) FROM PT(n)

7.  SP = T(n) − T(n−1)
8.  for each t∈SP do
9.      rid = rid + 1
10.     insert tuple (t[QI1],t[QI2],...,t[QIm],rid) into QIT(n)
11.     insert tuple (rid,t[S],1/m) into PT(n)

    // add random noise
12.  for i = 1 to m−1 do
13.     value = randomly pick a non-duplicated value from the domain of S
14.     insert tuple (rid,value,1/m) into PT(n)

15. return QIT(n) and PT(n)
```

[그림 1] QIT-PT 익명화 알고리즘 pseudo code

m-invariant 일반화 기법은 공격자가 어떤 개인의 민감한 속성 값을 미리 알고 있는 경우에는 심각한 프라이버시 침해 전파 문제가 발생한다. 공격자가 어떤 $t_1 \in T(i)$ $(1 \le i \le n)$의 민감한 속성을 알고 있다고 가정하자. t_1가 포함된 모든 $T(x),...,T(y)$ $(1 \le x < y \le n)$에서 t_1와 같은 동등클래스에 속한 레코드들의 집합을 V라 하자. t_1의 민감한 속성이 노출됨으로 인해 $t_2 \in V$의 시그니처에서 원래 시그니처에서 t_1의 민감한 속성을 제외하게 되고 공격자는 t_2의 민감한 속성을 최대 $1/(m-1)$의 확률로 알 수 있다. 영향은 여기서 그치지 않고 t_2과 같은 동등클래스에 속한 또 다른 레코드들에게 까지 전파될 수 있다.

제안 기법은 각 일반화를 사용하지 않고 각 레코드마다 민감한 속성을 따로 저장하여 레코드들 간의 민감한 속성 값에 관한 연관 관계를 제거하였기 때문에 프라이버시 침해의 전파 문제가 발생하지 않는다.

3.2.2 데이터 품질

익명화 알고리즘은 기본적으로 두 가지 목적을 가진다. 프라이버시 침해를 방지하여야 하고 통계 분석을 위해 정확한 통계 정보를 제공해야 한다. 배포된 마이크로데이터는 주로 집합 질의에 사용되며 데이터의 품질은

익명화된 데이블의 수행 결과와 원본 데이터의 질의 수행 결과 사이의 오차가 작을수록 높다. 본 절에서는 이전 연구[17]에서 데이터의 품질을 평가하기 위해 사용됐던 다음과 같은 COUNT 쿼리를 제안 기법을 사용해 익명화된 테이블에 수행한 결과에 대해 분석한다.

Q2: SELECT COUNT(*) FROM Microdata
WHERE pred(QI_1) AND ... AND
pred(QI_m) AND pred(S)

pred(S)는 민감한 속성 S에 대한 조건이고 pred(QI_i)는 속성 QI_i에 대한 조건이며 다음과 같은 형태를 가진다.

$QI_i \in (-\infty, \infty)$ 또는 $QI_i \in [x_i, y_i]$

x_i와 y_i는 QI_i의 도메인에 속한 값이다. 예를 들어 [표 1]과 같은 마이크로데이터를 대상으로 하면 pred(QI_1)은 $Age \in (-\infty, 30)$, pred(QI_2)는 $ZIP \in [10001, 20000]$ $ZIP \in [10001, 20000]$, pred(S)는 $Disease =$ '간염'이 될 수 있다.

익명화된 테이블 QIT와 PT를 대상으로 COUNT 쿼리를 수행한 결과는 다음과 같이 계산한다. QIT와 PT를 조인하면 다음과 같은 스키마를 가지는 테이블을 얻을 수 있다.

$(QI_1, QI_2, \ldots, QI_m, Row\_ID, S, PROB)$

Q2와 같은 쿼리는 QIT$\triangleright\triangleleft$PT에서 다음과 같이 해석된다. 즉, pred(QI_1), ..., pred(QI_m)과 pred(S)를 만족하는 레코드들의 $PROB$ 컬럼의 합을 구하면 익명화된 테이블에서 COUNT 쿼리의 결과 값을 구할 수 있다.

Q3: SELECT SUM(PROB) FROM QIT$\triangleright\triangleleft$PT
WHERE pred(QI_1) AND ... AND
pred(QI_m) AND pred(S)

Q2과 같은 형태의 쿼리를 제안 기법으로 익명화된 테이블을 대상으로 수행했을 때 받을 수 있는 기댓값(expectation)은 다음과 같다. 원본 테이블 T에서 pred(QI_1), ..., pred(QI_m)을 만족하는 레코드들의 집합을 R_T, pred(QI_1), ..., pred(QI_m)과 pred(S)를 만족하는 레코드들의 집합을 RS_T라 하자. QIT$\triangleright\triangleleft$PT에서 각 $t \in R_T$에 대응하는 레코드들의 집합을 R_{QIT-PT}, $t \in RS_T$에 대응하는 레코드들의 집합을 RS_{QIT-PT}라 하자. 원본테이블의 쿼리 결과는 $|RS_T|$이다. 알고리즘에 의해 RS_{QIT-PT}에 속하는 레코드들은 조건을 만족하는 민감한 속성 값을 $1/m$의 확률로 가지고 있다. $R_T - RS_T$에 속하는 레코드들이 QIT$\triangleright\triangleleft$PT에서 pred($S$)를 만족하는 S 값을 가질 확률은 $P(S) = \sum_{k=1}^{m-1} \frac{1}{|D|-k}$이다. D는 민감한 속성 S의 도메인을 의미한다. 따라서 QIT-PT 익명화 기법으로 익명화된 테이블에서 COUNT 쿼리의 기댓값은 다음과 같다.

$$E = \frac{|RS_T| + P(S)|R_T - RS_T|}{m} \quad (1)$$

원본 테이블과 익명화된 테이블의 쿼리 결과 오류는 다음과 같다.

$Err = \|RS_T| - E|$

식 (1)에서 $0 \le |R_T| \le |T|$ 이고 $0 \le |R_T - RS_T| \le |R_T|$ 이므로 $0 \le Err \le |T|\frac{m-1}{m}$이다.

Ⅳ. 결 론

본 논문에서는 삽입/삭제가 일어나는 동적 데이터베이스 환경에서 데이터를 재배포함으로써 발생하는 프라이버시 침해 문제를 해결할 수 있는 알고리즘을 제안하였다. 제안 기법은 m-invariant와 동등한 프라이버시 보호 수준을 제공한다. 일반화 대신 민감한 속성 값에 노이즈를 추가하는 기법으로 일반화로 인한 정보의 손실이 발생하지 않아 데이터의 품질을 높일 수 있는 기법이다. 또한 공격자가 어떤 개인의 민감한 속성을 이미 알고 있을 때 그 영향이 다른 레코드에 전파되는 것을 방지한다.

본 논문에서 제안한 기법은 범주형의 민감한 속성을 가지는 데이터일 경우에만 적용할 수 있다는 것이 단점이다. 이를 보완하기 위해 향후 연구 과제로 수치 형식의 민감한 속성을 가지는 마이크로데이터를 안전하게 익명화하면서 COUNT 쿼리 외에 SUM, MIN, MAX

등의 쿼리 결과의 정확성을 높일 수 있는 기법, 일반화를 함께 사용해 개인이 데이터에 포함되어 있다는 사실을 숨길 수 있는 기법, 두 개 이상의 민감한 속성을 가지는 마이크로데이터의 익명화 기법에 대한 연구를 수행할 계획이다.

참고문헌

[1] J.-W. Byun, Y. Sohn, E.Bertino, and N. Li, “Secure anonymization for incremental datasets” In Proceedings of the 3rd VLDB Workshop on Secure Data Management, pages 48-63, 2006.

[2] B. C. M. Fung, K. Wang, P. S. Yu, “Top- Down Specialization for Information and Privacy Preservation”, In International Conference on Data Engineering (ICDE), pp. 205-216, 2005.

[3] B. C. M. Fung, Ke Wang, A. W.-C. Fu, J. Pei, “Anonymity for Continuous Data Publishing”, In Proceedings of International Conference on Extending Database Technology (EDBT), 2008.

[4] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, “Fast data anonymization with low information loss”, In Proceedings of Very Large Data Bases (VLDB), pp. 758-769, 2007.

[5] K. LeFevre, D. J. Dewitt, and R. Ramakrishnan, “Incognito: Effective full-domain k-anonymity”, In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 49-60, 2005.

[6] K. LeFevre, D. J. DeWitt, R. Ramakrishnan, “Mondrian Multidimensional K-anonymity”, In International Conference on Data Engineering (ICDE), pp. 25-25, 2006.

[7] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity”, In International Conference on Data Engineering (ICDE), pp. 106-115, 2007.

[8] A. Machanavajjhala, J. Gehrke, and D. Kifer, “l-Diversity: Privacy beyond k-anonymity”. In International Conference on Data Engineering (ICDE), pp. 24-24, 2006.

[9] A. Meyerson and R. Williams, “On the complexity of optimal k-anonymity”, In Proceedings the ACM SIGMOD-SIGACT- SIGART Principles of Database Systems. pp. 223-228, 2004.

[10] H. Park, K. Shim, “Approximate Algorithms for k-Anonymity”, In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 67-78, 2007.

[11] P. Samarati, “Protecting respondents’ identities in microdata release”, IEEE Transactions on Knowledge and Data Engineering, 13(6), pp. 1010-1027, 2001.

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

[13] L. Sweeney, “Achieving k-anonymity privacy protection using generalization and suppression”, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), pp. 571-588, 2002.

[14] K. Wang, B. C. M. Fung, “Anonymizing Sequential Releases”, In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 414-423, 2006.

[15] X. Xiao and Y. Tao, “Anatomy: Simple and effective privacy preservation”, In Proceedings of Very Large Data Bases (VLDB), pp. 139-150, 2006.

[16] X. Xiao and Y. Tao. “m-Invariance: Towards privacy preserving re-publication of dynamic datasets”, In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 689-700, 2007.

[17] X. Xiao and Y. Tao. “Dynamic Anonymization: Accurate Statistical Analysis with Privacy Preservation”, In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 2008.

[18] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, “Aggregate query answering on anony-

mized tables", In International Conference on Data Engineering (ICDE), pp. 116-125, 2007.
[19] 변창우, 김재환, 이향진, 강연정, 박석, "안전한 데이터베이스 환경에서 삭제 시 효과적인 데이터 익명화 유지 기법", 한국정보보호학회논문지, v.17, no.3, pp. 69-80, 2007.

<著 者 紹 介>

이 주 창 (Joochang Lee) 학생회원
2007년 : 성균관대학교 정보통신공학부 학사
2007년~현재 : 성균관대학교 전자전기컴퓨터공학과 석사과정
<관심분야> 데이터베이스 보안, 데이터 마이닝

안 성 준 (Sung Jun Ahn)
1985년: 서울대 기계설계학과 학사
1987년: 한국과학기술원 생산공학과 석사
2004년: University of Stuttgart 박사
1985년~1990년: 금성사 가전연구소 주임연구원
1990년~2004년: Fraunhofer IPA(독) 연구원
2005년~현재: 성균관대학교 정보통신공학부 조교수
<관심분야> 공간정보처리, 패턴인식

원 동 호 (Dong ho Won) 종신회원
1976년~1988년 : 성균관대학교 전자공학과 (학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구원 전임연구원
1985년~1986년 : 일본 동경공업대 객원연구원
1988년~2003년 : 성균관대학교 교학처장, 전기전자및컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
2002년~2003년 : 한국정보보호학회장
현재 : 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정보통신부지정 정보보호인증기술연구센터장
<관심분야> 암호이론, 정보이론, 정보보호

김 응 모 (Ung Mo Kim) 정회원
1981년 : 성균관대학교 수학과 학사
1986년 : Old Dominion University 컴퓨터 과학 석사
1990년 : Northwestern University 컴퓨터 과학 박사
1997년 : 외무부 평가위원
1997년~1998년 : 한국정보처리학회 논문지 편집위원
1999년~2001년 : 한국정보처리학회 논문지 편집부위원장
2000년~2001년 : 대한상공회의소 전문위원
1994년~현재 : 성균관대학교 정보통신공학부 교수
<관심분야> 데이터베이스 보안, 데이터 마이닝, 정보보호