Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.6, Dec. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.6.1079

다자 간 환경에서 수직 분할된 데이터에서 프라이버시 보존 k번째 항목의 score 계산

홍 준 희,† 정 재 열, 정 익 래‡
고려대학교 정보보호대학원

Privacy-Preserving Kth Element Score over Vertically Partitioned Data on Multi-Party

Jun Hee Hong,† Jay Yeol Jung, Ik Rae Jeong‡
Graduate School of Information Management and Security, Korea University

요 약

데이터 마이닝은 보유한 데이터를 가공하여 패턴 분석이나 마케팅 등에 활용할 수 있는 유용한 정보를 얻어내는 기술이다. 그러나 이러한 데이터 마이닝 기술을 사용 시 데이터의 제공자의 개인정보가 의도치 않게 유출 될 가능성이 존재하게 된다. 이러한 정보의 유출을 막기 위하여 여러 가지 프라이버시를 보호하는 기법이 연구되고 있다.

수직 분할 데이터는 같은 집단에 관한 데이터가 복수의 소유자에게 나누어 제공되어 있는 상태를 말한다. 이러한 수직 분할된 데이터에서 프라이버시를 보호하면서 k번째 항목과 $(k+1)$ 번째 항목을 $score$값을 이용하여 구분하는 방법이 개발되었다. 그러나 기존의 연구에서는 양자간의 환경에서만 사용이 가능하였기 때문에 본 논문에서는 Paillier 암호화 기법을 사용하여, 기존의 연구를 다자간 환경으로 확장한 기법을 제안한다.

ABSTRACT

Data mining is a technique to get the useful information that can be utilized for marketing and pattern analysis by processing the data that we have. However, when we use this technique, data provider's personal data can be leaked by accident. To protect these data from leakage, there were several techniques have been studied to preserve privacy.

Vertically partitioned data is a state called that the data is separately provided to various number of user. On these vertically partitioned data, there was some methods developed to distinguishing kth element and $(k+1)$ th element by using $score$. However, in previous method, we can only use on two-party case, so in this paper, we propose the extended technique by using paillier cryptosystem which can use on multi-party case.

Keywords: Privacy preserving, Secure multiparty computation, Vertically partitoned data

I. 서 론

현대 사회는 기초적인 정보 획득뿐만 아니라 온라인 쇼핑몰이나 소셜 커머스 등을 이용한 전자상거래, 인터넷뱅킹, 다이렉트 보험 및 온라인 주식거래 등 일상생활에 필요한 것의 대부분을 온라인에 의지하고 있다. 이러한 온라인상에 존재하는 사용자들의 활동 정보를 모아서 이를 마케팅에 활용하는 기법을 데이터 마이닝이라고 한다.

현대 사회에서 데이터 마이닝은 기업에서 마케팅

접수일(2014년 9월 25일), 수정일(1차: 2014년 10월 29일, 2차: 2014년 10월 31일), 게재확정일(2014년 10월 31일)
† 주저자, juni8721@nate.com
‡ 교신저자, irjeong@korea.ac.kr(Corresponding author)

업무를 수행함에 있어 가장 필수적인 요소가 되었다. 이로 인해 기업의 입장에서는 매우 경제적이고 효율적인 마케팅을 할 수 있게 되었다. 그러나 데이터 마이닝은 기본적으로 해당 사용자의 정보를 마케팅에 이용하는 것으로 개인정보가 악의적인 사용자에게 의도치 않게 노출될 가능성이 존재하게 된다. 이로 인하여 사용자의 개인정보가 무분별하게 사용되어 개인정보 유출의 1차적인 피해뿐만 아니라, 개인정보의 재판매로 인한 스팸메일, 대출사기 문자 등이 발송되는 피해가 발생할 가능성이 매우 높아지게 된다. 또한 보이스피싱 등으로 인하여 금전적 피해가 발생하게 될 가능성이 매우 높기 때문에 이러한 피해를 줄이기 위해 개인정보를 보호하면서 데이터 마이닝 기술을 적용할 수 있는 여러 기법들이 개발되고, 이에 대한 안전성에 관하여 많은 연구가 진행되고 있다.

수직 분할 데이터는 최근 연구되고 있는 데이터 형태 중의 하나로서, 최근 현대 사회가 다양하게 연결됨으로 인하여 하나의 개인정보가 여러 기관에 제공되는 것에 착안하여 만들어진 기법이다[8]. 성적의 경우를 예를 들 경우, 한 학년의 국어 과목 담당자와 수학 과목 담당자가 각각 자신이 담당하는 과목의 데이터는 가지고 있으나, 다른 과목의 데이터는 가지고 있지 않기 때문에, 각각 국어 점수와 수학 점수는 수직 분할된 데이터라고 할 수 있다. 이러한 수직 분할 데이터를 이용하여 데이터 마이닝을 수행할 경우 기존에 존재하는 데이터 마이닝 기법을 사용할 경우에 비하여 좀 더 효율적인 마케팅 전략을 세울 수 있으며, 이를 통해 명확한 소비자 집단을 타게팅하여 마케팅을 진행할 수 있기 때문에 비용 감소뿐만 아니라 효과적인 마케팅으로 인한 매출 상승 등의 효과 역시 누릴 수 있다.

그러나 수직 분할 데이터에 사용되는 개인정보를 아무런 안전장치 없이 사용하였을 때, 이 데이터가 인사고과 점수나 시험 성적 등 민감한 개인정보를 포함하고 있다면, 데이터가 노출되었을 경우 심각한 프라이버시 침해를 야기할 수 있기 때문에 이를 방지할 수 있는 안전장치가 필요하다.

본 논문에서는 이러한 위험성을 사전에 방지하고자 기존의 양자간 환경에서 수직 분할된 데이터에서 사용자의 개인정보의 노출을 최소화할 수 있는 기법을 다자간으로 확장한 기법을 제안한다. 기존의 수직 분할된 데이터에 관한 연구에서는 양자 간의 수직 분할된 데이터에서의 프라이버시 보호를 위하여 직접적으로 원하는 값을 노출하지 않는 $score$값을 사용하였다. 본 논문에서는 기존의 연구와 마찬가지로 $score$값을 사용하여 원하는 값이 일정한 범위 안에 존재하여 직접적으로 노출되지 않도록 하였으며, 기존의 연구에서와는 달리 다자간 환경으로의 확장이 가능하도록 기법을 설계하였다.

본 논문에서는 RSA 암호화 방식을 기반으로 한 Paillier 암호화 방식을 적용하여 양자 간의 환경에서만 사용 가능한 것이 아니라 다자간의 수직 분할된 환경에서도 사용할 수 있도록 해당 기법을 개선한 방식을 제안한다. 또한 Paillier 암호화 방식을 적용함에 따라 기존의 논문과 비교하였을 때에도 안전성을 보장할 수 있다.

II. 배경 지식

2.1 A. Yao의 연구[1]

Yao의 백만장자 문제는 안전하게 양자간 통신을 하고자 하는 것으로서, A. Yao에 의하여 제안된 방법이다.

1982년 A. Yao의 연구에 의하여 처음 제안된 이 문제는 두 명의 백만장자가 서로의 재산에 관한 정보를 노출하지 않고 상대방과 자신의 재산을 비교하고자 하는 것에서 출발하였다. 예를 들어, Alice와 Bob이라는 두 명의 백만장자가 있을 때, 두 명의 백만장자의 재산을 비교하기 위한 기법이다. Alice와 Bob은 RSA 공개키(e,n)과 (d,n)을 가지고 있다고 가정하고, 두 백만장자 재산의 최고자리의 숫자를 각각 I와 J라고 가정한 뒤, Fig. 1.과 같은 순서로 프로토콜을 진행한다. 먼저 Bob에서 임의의 자연수 x를 선택한 뒤 암호화하여 암호문 C를 생성한다. 생성된 암호문 C를 이용하여 아래의 식 (1)과 같이 메시지 m을 생성하여 Alice에게 전송한다.

$$C - J + 1 = m \bmod n \tag{1}$$

Alice는 Bob에게서 전송받은 메시지 m을 i를 이용하여 복호화하여 Y_i를 계산한다. 이 때, 복호화한 Y_i에서 $i \in [1,10]$일 경우 $Y_J = x$ 로 하고, 복호화한 Y_i를 p로 나눈 나머지를 Z_i로 한다.

이 때, $i > I$일 경우 W_i는 $Z_i + 1$을 p로 나눈 나머지로 생성한다. 생성된 $W_1, W_2, \ldots, W_{10}$까지의 값과 p를 Bob에게 다시 전송한다. Bob은 넘겨받은 W_J의 값을 p로 나눈 나머지가 x일 경우 Alice의

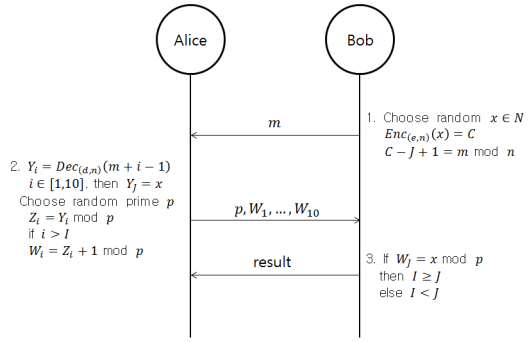

Fig. 1. Protocol of Yao's millionaire problem

재산이 같거나 크다는 것을 의미하며, 그 외의 경우 Bob의 재산이 크다는 것을 의미한다.

2.2 수직 분할 데이터[8]

데이터를 사용하는 여러 사용자가 동일한 집단의 데이터에 대하여 서로 다른 속성의 정보를 가지고 있을 때 이 데이터는 수직 분할되었다고 할 수 있다. 예를 들어 동일한 집합에 대하여 병원에서 보유하고 있는 정보와 무선통신 사업자가 보유하고 있는 정보가 있다고 가정할 경우, 병원에서는 해당 집합의 구성원들의 병원 진료 기록과 병력 기록을 수집한다. 또한 무선통신 사업자의 경우 해당 집합의 사용자가 사용하는 휴대폰의 종류와 배터리의 종류, 매일 통화하는 대략적인 시간 등의 정보를 수집할 수 있다. 이와 같이 서로 다른 두 가지 정보를 이용하여 데이터마이닝 알고리즘을 수행할 경우 해당 고객의 생활 패턴이나 두 사업자 외의 기타 사업자와의 거래 내역 등 전혀 예상치 못한 개인정보가 노출될 가능성이 있다. 이렇듯 사용자가 원하지 않는 개인정보가 노출되어 악의적인 공격자에 의하여 사용될 경우, 사용자에게 피해를 입힐 가능성이 매우 높기 때문에 이러한 경우 역시 프라이버시 보호가 필요하다.

본 논문에서 제시하는 기법은 수직 분할된 데이터 중 연산이 가능한 수치화된 데이터를 가지고 데이터마이닝을 수행할 경우에 사용할 수 있는 기법이다. 즉, 학생의 총점을 이용한 상위 그룹과 하위 그룹을 나누거나, 인사 고과 점수를 이용하여 승진 대상자를 나누는 등 수치화 된 데이터를 대상으로 하여 연산을 수행한다.

2.3 Paillier 암호화 방식[7]

Paillier 암호화 기법은 1999년 P. Paillier에 의하여 제시된 암호화 기법이다. 기존의 RSA와 유사한 기법으로 큰 소수 p, q를 이용하여 생성한 n과 g를 공개키로 사용하고, λ와 μ를 비밀키로 사용하며, 준동형 성질을 만족한다. 본 논문 역시 이 방식을 이용하여 암호화 및 복호화를 수행한다.

Paillier 암호화 기법은 Table 1.과 같이 충분히 큰 소수 p, q를 선택하는 것에서 시작한다. 선택한 p, q가 $\gcd(pq, (p-1)(q-1)) = 1$ 을 만족하는지 확인한 후, 만약 만족하지 않을 경우 p, q를 다시 선택한다. 선택한 p, q가 테스트를 통과할 경우, p, q를 이용하여 n과 λ를 계산한다. p, q를 곱하여 n을 계산하고, $p-1$과 $q-1$의 최소공배수를 계산하여 λ로 한다. 또한 계산한 n을 이용하여 $g = n+1$을 만족하는 g를 생성한 뒤, 함수 L을 (2)과 같이 정의한다. 이 때, $L(u)$는 $u-1$의 값을 n으로 나눈 몫이 된다. μ는 앞서 생성한 g와 함수 L을 이용하여 (3)와 같이 계산한다. 이와 같이 생성된 n과 g를 공개키로 사용하고 λ와 μ를 비밀키로 사용한다.

암호화 과정에서는 키 생성 과정에 의해 생성된 키를 이용하여 Z_n의 원소 중 선택한 메시지를 공개키 g에 평문 메시지 m을 제곱하고, Z_n^*의 원소 중에서 임의로 선택한 r에 공개키 n을 제곱한 값을 곱하여 n^2으로 나눈 나머지를 최종적인 암호문으로 한다.

복호화 과정에서는 암호화 과정을 통해 암호화된 값 c를 (2)에서 정의한 함수 L과 키 생성 과정에서 생성한 μ를 이용하여 복호화한다. 암호화된 값 c를 λ제곱하여 n^2으로 나눈 나머지를 (2)의 함수 L에 대입한 뒤, 계산한 값에 μ를 곱하여 n으로 나눈 나머지를 계산하면 원래의 평문 m을 계산할 수 있다.

III. 관련 연구

공개키 암호에 관한 연구는 1976년 W. Diffie와 E. Hellman[18]이 제안한 안전하지 않은 통신망에서의 키 교환을 위한 알고리즘에서 시작되었으며, 이산대수 문제의 어려움에 기반하여 만들어진 알고리즘이다. 이후, 해당 연구를 바탕으로 ElGamal 암호[19], RSA 암호[20], Rabin 암호[21] 등의 공개키 암호가 만들어지게 되었다. 1999년 P. Paillier[7]의 연구에 의하여 제안된 암호화 방식은 RSA를 이용

한 암호화 방식으로, 큰 소수 p, q를 이용하여 준동형(homomorphic) 성질을 이용하여 다자간 환경에서의 연산을 수행할 수 있도록 알고리즘을 설계하였다.

양자간 환경에서의 안전한 연산 방법에 관한 연구는 1982년 A. Yao[1]에 의하여 최초로 제안되었다. 해당 연구는 두 명의 백만장자가 누가 더 부자인지 자신의 재산을 노출하지 않고 비교하고자 하는 것에서 시작되었다. 그러나 A. Yao의 연구에 의하여 제안된 알고리즘은 양자 간의 환경을 기반으로 사용하도록 알고리즘이 구성되어 있어 다수의 사용자가 있는 경우 사용이 어려운 단점이 있었다. 이에 1998년 O. Golereich[3]의 연구에 의하여 기존의 A. Yao에 의하여 제안된 양자 간의 안전한 연산 방식을 다자간의 환경에서 적용 가능하도록 확장한 방식이 제안되었고, 이를 바탕으로 여러 가지 SMC(Secure Multiparty Computation) 기법에 대한 연구가 진행되었다. 2003년 R. Agrawal 등[12]의 연구에서 개인 DB에서 정보를 공유할 수 있는 기법이 제안되었다. 그러나 해당 기법에서는 프라이버시를 보호할 수 있는 안전장치가 마련되어 있지 않아 이를 보완하기 위하여 프라이버시를 보장하는 데이터 마이닝 기법, k개의 익명 클러스터링 기법[13], 온라인 상에서의 분석 작업 수행 시 프라이버시 보장 기법[14] 등에 관한 연구가 진행되었다. 이러한 프라이버시 보장 기법은 기존의 연구들에 비하여 사용자의 정보가 노출되는 것이 많이 줄어들었으나, 여전히 데이터 마이닝 수행 시 일부 사용자의 정보가 직접적으로 노출될 위험성을 가지고 있었다.

Table 1. Technique of Paillier cryptosystem

Keygen(1^{λ})

Choose random prime number p, q

$\gcd(pq, (p-1)(q-1)) = 1$

Compute n, λ

$n = pq$, $\lambda = lcm(p-1, q-1)$

Select $g = n+1$

$$L(u) = \frac{u-1}{n} \quad (2)$$

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n \quad (3)$$

Public key : (n, g)

Private key : (λ, μ)

Enc(m, n, g)

Select Random $r \in Z_n^*$

Compute ciphertext $c = g^m r^n \bmod n^2$

Dec(c, n, λ, μ)

Compute plaintext

$m = L(c^{\lambda} \bmod n^2)\mu \bmod n$

프라이버시 보존 데이터 마이닝 기법에 관한 연구는 2000년 R. Agrawal와 R. Srikant[16]에 의하여 기존의 데이터 마이닝 기법을 수행하면서 노출될 수 있는 사용자의 프라이버시를 보호하고자 하는 기법이다. 해당 논문 이전의 데이터 마이닝 기법들은 사용자의 안전보다 효율성에 중점을 두고 연구가 진행되어 왔으나, 개인정보에 관한 관심이 점차 높아지면서 관련 연구의 필요성이 대두되었다. 이를 바탕으로 2002년 J. Vaidya와 C. Clifton[4]의 연구에 의하여 수직 분할된 데이터에 대한 개념이 정의되었고, 이후 수직 분할된 데이터에서 데이터 마이닝을 수행할 경우, 프라이버시를 보장할 수 있는 방법에 관한 연구가 시작되었다. 또한 수직 분할된 데이터뿐만 아니라 분산된 데이터에서의 프라이버시 보장 데이터 마이닝, 수직 분할된 데이터에서의 k 군집화[17], 수직 분할된 데이터 관리[9] 등 여러 방면으로의 연구가 진행되었다. 이후 2009년 J. Vaidya와 C. Clifton[5]의 연구에서 제안된 양자간 환경에서 수직 분할된 데이터에서 k번째 원소의 $score$값을 찾는 방식이 제안되었으며, 이를 이용하여 해당 기법은 개인정보의 노출을 최소화하면서 연산을 수행하여 의미있는 값을 계산할 수 있다. 그러나, 해당 연구에서는 Yao 비교법을 사용하여 양자 간의 환경에서만 사용이 가능하다는 단점이 존재한다.

3.1 J. Vaidya와 C. Clifton의 연구[5]

J. Vaidya와 C. Clifton의 연구에서는 양자 간 환경에서 수직 분할된 집단의 데이터에서 사용자의 정보를 노출하지 않고, k번째 값과 $(k+1)$번째 값을 구별하는 역할을 하는 $score$값을 구하는 방식을 제안한다. 제안하는 방식은 이진 검색을 이용하여 $score$값이 존재 할 수 있는 범위를 줄여 최종적인 $score$값을 구하는 방식이다.

해당 연구는 각각의 정보가 서로 다른 곳에 분리되어 있고, 수직 분할되어 있는 데이터에서 최상위의 k개와 나머지를 구분하는 문제를 분산 처리하는 기법을 제안한다. 또한 모든 데이터를 이용해야 원하는 결과값을 도출해 낼 수 있게 함으로서, 분리된 정보를 가진 모든 곳에서 동등한 권한을 가질 수 있도록

하였다. 이를 위해 Table 2.의 알고리즘과 같이 Yao 비교법을 이용하여 성공적으로 두 개의 데이터의 덧셈 연산을 수행하였다.

Table 2.의 $lbound$와 $ubound$는 이진 검색을 수행하기 위한 것으로, $lbound$는 0으로, $ubound$는 A의 값과 B의 값을 더하여 나올 수 있는 최대값인 $|F|$ 로 초기값을 설정한 뒤 연산을 수행한다. $estimate$를 $lbound$와 $ubound$의 중간값으로 하고, 이진 검색을 통해 범위를 줄여주는데 사용한다. 이후 Yao 비교법을 이용하여 연산에 사용되는 A와 B의 각 원소 a_i와 b_i를 더한 값이 $estimate$보다 작거나 같을 경우, la_i와 lb_i를 더한 값이 $|F|$ 로 나눈 나머지가 1이 되도록 la_i와 lb_i를 정하여 각각 A와 B에게 돌려준다. 만약 그렇지 않을 경우, la_i와 lb_i를 더한 값이 $|F|$ 로 나눈 나머지가 0이 되도록 la_i와 lb_i를 정하여 돌려준다. 해당 연구 이후 비밀 공유 기법을 이용한 프라이버시 보장 top-k 문제, 분산 네트워크에서의 프라이버시 보장 문제 등으로 연구가 진행되어 왔으나 해당 기법을 다자간으로 확장하는 방법에 관한 연구는 이루어지지 않아 본 논문에서는 RSA 기반의 Paillier 암호화 방식을 적용하여 해당 기법을 확장하였다.

Yao 비교법을 이용한 모든 과정이 끝난 후 A에서는 결과값으로 출력된 la_i를 모두 더하여 $|F|$ 로 나누었을 때의 나머지인 ll_a를 생성하고, B에서도 A와 마찬가지로 모든 lb_i를 더하여 $|F|$ 로 나누었을 때의 나머지인 ll_b를 생성한다. 이렇게 생성한 ll_a와 ll_b를 더하여 $|F|$ 로 나눈 나머지가 k보다 클 경우, $ubound$를 $estimate$값으로 재설정하고 다시 알고리즘을 수행한다. 만약 ll_a와 ll_b를 더하여 $|F|$ 로 나눈 나머지가 k보다 작을 경우, $lbound$를 $estimate$값으로 재설정하고, ll_a와 ll_b를 더한 값이 k와 같을 경우 $estimate$를 $score$로 하고 알고리즘을 종료한다. 만약 $ubound - lbound = 1$ 일 경우, ll_a와 ll_b를 더하여 $|F|$ 로 나눈 나머지가 k보다 클 경우 더 이상 알고리즘의 수행이 불가능하므로 $lbound$를 $score$값으로 출력하고 알고리즘을 종료한다. 마찬가지 방법으로 ll_a와 ll_b를 더한 값이 k보다 작을 경우, $ubound$를 $score$값으로 출력하고 알고리즘을 종료한다.

Table 2. Algorithm of distinguishing kth element over two party

A and B shares $a_1, a_2, \dots, a_n$, and $b_1, b_2, \dots, b_n$

$lbound \leftarrow 0$

$ubound \leftarrow |F|$

while true do

$estimate \leftarrow \left\lceil \frac{lbound + ubound}{2} \right\rceil$

for $i = 1$ to n do

Use Yao's secure comparison with input a_i, b_i and output la_i, lb_i such that

$la_i + lb_i = 1 \pmod{|F|}$ if $a_i + b_i \leq estimate \pmod{|F|}$, otherwise

$la_i + lb_i = 0 \pmod{|F|}$

end for

A : $ll_a \leftarrow \sum_{i=1}^{n} la_i \pmod{|F|}$

B : $ll_b \leftarrow \sum_{i=1}^{n} lb_i \pmod{|F|}$

if $ll_a + ll_b \pmod{|F|} > k$ then

$ubound \leftarrow estimate$

if $ubound - lbound = 1$ then

return $lbound$

end if

else if $ll_a + ll_b \pmod{|F|} < k$ then

$lbound \leftarrow estimate$

if $ubound - lbound = 1$ then

return $ubound$

end if

else

return $estimate$

end if

end while

IV. 제안하는 기법

본 장에서는 Paillier 암호화 방식을 적용하여 다자간의 연산이 가능하도록 한 기본 기법과 기본 기법을 사용하였을 때 발생할 수 있는 문제점을 임의의 난수를 추가하고, 특정 사용자에게 권한이 집중되지않도록 개선한 방식을 제안한다. 제안하는 기법은 기존의

양자간의 합의 크기를 비교하는 방식을 확장하여 다자간의 합의 크기를 안전하게 비교하기 위하여 Paillier 암호화 방식을 적용하였고, 임의의 난수를 연산에 추가로 사용하여 좀 더 안전한 방식으로 개선하였다. 기존의 J. Vaidya 등의 연구에서 사용한 방식은 2명의 사용자가 존재할 경우 3명 이상의 사용자로 확장하는 경우 Yao의 안전한 비교법을 사용하였기 때문에 연산 과정에서 중간 합산 결과가 최종 연산 수행자에게 정보가 노출되기 때문에 3인 이상의 다자간 연산으로는 확장할 수 없는 문제점이 있었다.

4.1 Paillier 암호화를 적용한 다자간 연산의 가본 기법

제안하는 기법은 Fig. 2와 같이 P_a의 공개키를 가지고 P_a의 원소를 암호화한 뒤에 P_b로 전송한다. P_b에서는 P_a에서 전송받은 값에 P_b의 원소를 P_a의 공개키로 암호화한 값을 곱하여 P_c에 전송한다. P_c에서는 P_b에서와 마찬가지로 P_c의 원소를 P_a의 공개키로 암호화한 값을 P_b에서 전송받은 값에 곱하고, 복호화 권한이 있는 P_a에게 추가적인 정보를 제공하지 않기 위해서 $-estimate$값에 해당하는 $mean$값을 P_a의 공개키로 암호화하여 곱하여 준 뒤 P_a에게 전송한다.

P_a에서는 P_c로부터 전달받은 값을 복호화하게 되는데 이때, Paillier의 준동형 성질을 이용하게 된다. 준동형 성질에 의하여 각 성분의 값을 암호화하여 곱한 값은 각각의 성분의 값을 더하여 암호화한 값과 같게 된다. 즉, P_a가 P_c로부터 전달받은 값인 $E_a(a_i)E_a(b_i)E_a(c_i)E_a(mean)$를 복호화할 경우 $E_a(a_i+b_i+c_i+mean)$ 을 복호화한 값과 동일한 결과값을 갖게 된다.

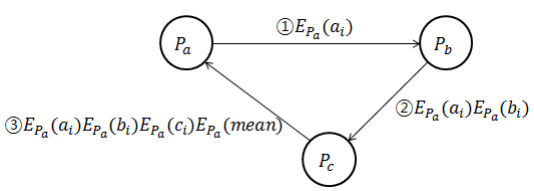

Fig. 2. flow chart of proposing encryption scheme

4.1.1 키 생성 알고리즘

Paillier 암호화 방식을 이용한 키 생성 알고리즘은 앞에서 언급한 Table 1.의 알고리즘을 사용한다. 본 기법에서는 Paillier 암호화 기법을 해당 연산에 적합하게 적용하기 위하여 아래의 식 (4)과 같은 조건을 추가한다.

$$n > 2^*|F| \tag{4}$$

식 (4)는 Paillier 암호화 기법을 사용하기 위해 도입한 것으로, 준동형 성질을 이용하여 암호문을 생성하여 판별 과정에서 처음 계산한 결과값과 비교하기 위하여 필요한 조건이다. 각 집합에서 연산하여 나올 수 있는 최대값인 $|F|$ 가 $estimate$가 가질 수 있는 최대값이다. 따라서 $score$값은 $|F|$ 와 $estimate$를 더한 값을 n으로 나눈 나머지가 0과의 대소 비교를 통하여 구하게 되므로 둘을 더한 값은 n보다 항상 작아야 한다.

4.1.2 기본 기법의 알고리즘 구성

해당 기법의 암호화 알고리즘은 Table 3.의 알고리즘과 같이 구성된다. P_a의 input값들의 집합을 D_a, P_b의 input값들의 집합을 D_b, P_c의 input값들의 집합을 D_c로 하고, 앞에서 언급한 암호화 기법에서 생성된 키를 이용하여 메시지를 공개키 g에 m제곱하고 Z_n^*에서 임의로 선택한 r에 n제곱한 뒤 n^2으로 나눈 나머지를 각 사용자의 암호문으로 한다.

Paillier 암호화 기법의 준동형의 성질을 이용하여 각 사용자가 암호화한 값을 복호화하지 않더라도 암호화된 값을 연산하여 원하는 값의 암호화 값을 구할 수 있으므로 암호문 생성 후 P_a에서 P_a의 공개키로 생성된 암호문 A을 P_b로 전송한다. P_b에서는 전송받은 P_a의 암호문 A에 P_a의 공개키를 이용하여 자신이 생성한 암호문 $E_a(b_1)$을 곱하여 n^2으로 나눈 나머지 B를 P_c에게 전송한다. 마지막으로 P_c에서 P_a의 공개키를 이용하여 자신이 생성한 암호문 $E_a(c_1)$와 $mean$을 P_a의 공개키를 이용하여 생성한 암호문 $E_a(mean)$을 계산한 뒤, B와 이 두 값을 곱하여 n^2으로 나눈 나머지를 계산하여 최종적인 암호문을 완성한다.

Paillier 암호화 방식은 암호화하여 곱한 값은 각각을 더하여 암호화 한 값과 동일하다. 따라서 이와 같은 준동형 성질을 이용하여 Table 3.에서와 같이 안전한 복호화를 수행한다.

암호화 과정에서 P_c에서 생성된 암호문을 비밀키를 가지고 있는 P_a에게 넘겨주어 암호문을 복호화한다. 준동형 성질에 의하여 각각의 원소를 암호화하여 곱한 것은 모든 원소를 더하여 암호화한 것과 같다. 즉, Table 3.에서와 같이 P_c에서 계산하는 값인 $E_a(a_j)E_a(b_j)E_a(c_j)E_a(mean) \bmod n$은 각 성분을 더하여 암호화한 $E_a(a_j+b_j+c_j+mean) \bmod n$ 과 같기 때문에 이러한 성질을 이용하여 암호화된 상태에서의 덧셈 연산이 가능하다.

복호화된 평문은 각 사용자의 성분의 합과 $mean$ 값의 합으로 복호화를 수행한 P_a는 사용자의 성분의 합을 알 수 없다.

복호화된 평문을 n으로 나눈 나머지를 $mean$값과 비교하여 복호화된 값이 클 경우 0을 P_c에게 전달하고, 반대의 경우 1을 최종 암호화 과정을 수행한 사용자 P_c에게 전달한다.

P_c는 P_a에서 전달된 값들을 전부 더하여 k와 비교한다. 만약 값들의 총합이 k보다 작을 경우 $estimate$보다 큰 값이 k개보다 적다는 것을 의미하므로 이진 연산을 이용하여 $ubound$의 값을 $estimate$값으로 바꾸어 같은 방식으로 암호화 및 복호화 알고리즘을 수행한다. 마찬가지로 값들의 총합이 k보다 큰 경우 $estimate$값보다 큰 값이 k개보다 많다는 것을 의미하므로 이 경우 $lbound$값을 $estimate$값으로 바꾸어 같은 방식으로 암호화와 복호화 알고리즘을 수행한다.

이와 같이 P_a에서 복호화 된 값들의 총합이 k를 만족할 때까지 알고리즘이 수행되며 이 때의 $estimate$값이 본 기법에서 구하고자 하는 $score$값이 된다.

4.2 Pailler 암호화를 적용한 개선된 다자간 연산 기법

앞에서 언급한 Paillier 암호화 방식을 적용한 기법은 $estimate$값을 그대로 사용함으로서 최종 연산자가 다른 사용자들의 값을 유추해낼 수 있어 정보의 노출이 발생할 가능성이 있었다. 제안하는 기법에서는 기본 기법에서 제안한 방식과 유사하며, Table

Table 3. The basic algorithm using Paillier cryptosystem

let P_a's dataset D_a, P_b's dataset D_b, and P_c's dataset D_c
$D_a = a_1, a_2, \ldots, a_m$, $D_b = b_1, b_2, \ldots, b_m$ and $D_c = c_1, c_2, \ldots, c_m$
$F = \{a_1+b_1+c_1, a_2+b_2+c_2, \ldots, a_m+b_m+c_m\} = \{f_1, f_2, \ldots, f_m\}$
$lbound \leftarrow 0$, $ubound \leftarrow |F|$
$estimate \leftarrow \left\lceil \frac{lbound + ubound}{2} \right\rceil$
$mean = -estimate \bmod n$
for i = 1 to m
select random $r_a, r_b, r_c \in Z_n^*$
P_a : compute $E_a(a_i) = g^{a_i} r_a^n \bmod n^2$
P_b : compute $E_a(b_i) = g^{b_i} r_b^n \bmod n^2$
P_c : compute $E_a(c_i) = g^{c_i} r_c^n \bmod n^2$ and $E_a(mean) = g^{mean} r_c^n \bmod n^2$
end for
for j = 1 to m
P_a : compute A$\leftarrow E_a(a_j)$
send A to P_b
P_b : compute B$\leftarrow E_a(a_j)E_a(b_j) \bmod n$
send B to P_c
P_c : compute C$\leftarrow E_a(a_j)E_a(b_j)E_a(c_j)E_a(mean_j) \bmod n$s
end C to P_a
P_a : decrypt C
if $(f_j + mean) \bmod n > mean$
$output_j = 0$
send $output_j$ to P_c
else
$output_j = 1$
send $output_j$ to P_c
end if
end for
P_c : if $\sum_{j=1}^{m} output_j > k$
$lbound \leftarrow estimate$
if $ubound - lbound = 1$ then
return $ubound$
end if
elseif $\sum_{j=1}^{m} output_j < k$
$ubound \leftarrow estimate$
if $ubound - lbound = 1$ then
return $lbound$
end if
else $\sum_{j=1}^{m} output_j = k$
return $estimate$
end if

1.의 키 생성 알고리즘을 사용한다. 기본 기법과는 달리 연산과정에 임의의 난수를 삽입하여 연산을 수행함으로 인하여 최종 연산자가 다른 사용자들의 값을 유추해낼 수 없도록 하였다. 또한 기본 기법에서 제안한 암호화 방식을 변형하여 사용자들의 공개키를 번갈아 사용하여 모든 사용자가 동등한 권한을 가질 수 있도록 구성하였다.

4.2.1 개선된 기법의 알고리즘 구성

암호화 알고리즘은 Table 1.에서 설명한 키 생성 방식에 의해 생성된 키를 이용하여 선택된 메시지를 공개키 g의 m제곱과 Z_n^*에서 임의로 선택한 r의 n제곱을 이용하여 평문 메시지 m을 암호화한다. Table 4.에서 알 수 있듯이 알고리즘의 수행 과정은 Table 3.에서의 방식과 유사하나, 일정 범위 내에서 임의의 값을 선택하는 방식을 추가하여 안전성을 높였고, 매 라운드 수행 시 서로 다른 사용자의 공개키를 사용하여 연산에 참여하는 사용자들이 동등한 권한을 가질 수 있도록 하였다. 제안하는 알고리즘은 앞서 5.1.2에서 제안한 기본 알고리즘과 마찬가지로 $lbound$의 값을 0, $ubound$의 값을 성분들의 합에서 나올 수 있는 최대값인 $|F|$ 로 설정한 뒤 두 값의 중간값인 $\left\lceil \frac{lbound+ubound}{2} \right\rceil$ 를 t로 설정한다. 그 다음 복호화 권한을 가진 사용자에게 다른 사용자들의 값을 노출하지 않기 위해 최종 암호화 수행자인 P_c에서 임의의 작은 값 α를 선택하고 t값을 중심으로 α의 범위 내에서 임의의 값을 선택하여 $estimate$값으로 한다. 이 때, α는 10 이내의 작은 숫자로 하되 알고리즘이 수행되는 횟수에 따라 이전 범위의 절반으로 점차 크기를 줄여 알고리즘 수행 과정에서 오류가 발생하지 않도록 한다. 이후 수행과정은 Table 3.에서 설명한 것과 같으며 알고리즘이 짝수번째 수행될 때에는 Table 4.에서의 두 번째 for문이 수행되는 부분을 Table 5.에서와 같이 변형하여 수행한다. 이때, 사용되는 공개키는 P_b의 공개키이므로 P_b에서 가장 먼저 값을 계산하여 P_a로 전송한 다음 P_a에서는 앞에서 수행한 방식과 마찬가지로 전송받은 값에 본인의 값을 곱하여 난수를 포함하여 연산하는 P_c에 전송한다. P_c에서는 연산을 수행한 뒤 값을 복호화할 수 있는 P_b에 값을 전송하는 방식으로 알고리즘을 수행한다. 이와 같은 방식으로

Table 4. The improved algorithm using Paillier cryptosystem(while performing odd times)

let P_a's dataset D_a, P_b's dataset D_b, and P_c's dataset D_c
$D_a = a_1, a_2, \dots, a_m$, $D_b = b_1, b_2, \dots, b_m$ and $D_c = c_1, c_2, \dots, c_m$
$F = \{a_1+b_1+c_1, a_2+b_2+c_2, \dots, a_m+b_m+c_m\} = \{f_1, f_2, \dots, f_m\}$
$lbound \leftarrow 0$, $ubound \leftarrow |F|$
$t = \left\lceil \frac{lbound+ubound}{2} \right\rceil$
P_c : select α, $estimate \in [t-\alpha, t+\alpha]$
$mean = -estimate \bmod n$
for i = 1 to m
  select random $r_a, r_b, r_c \in Z_n^*$
  P_a : compute $E_a(a_i) = g^{a_i} r_a^n \bmod n^2$
  P_b : compute $E_a(b_i) = g^{b_i} r_b^n \bmod n^2$
  P_c : compute $E_a(c_i) = g^{c_i} r_c^n \bmod n^2$ and $E_a(mean) = g^{mean} r_c^n \bmod n^2$
end for
for j = 1 to m
  P_a : compute A$\leftarrow E_a(a_j)$
    send A to P_b
  P_b : compute B$\leftarrow E_a(a_j)E_a(b_j) \bmod n$
    send B to P_c
  P_c : compute C$\leftarrow E_a(a_j)E_a(b_j)E_a(c_j)E_a(mean_j) \bmod n$s
    end C to P_a
  P_a : decrypt C
    if $(f_j + mean) \bmod n > mean$
      $output_j = 0$
      send $output_j$ to P_c
    else
      $output_j = 1$
      send $output_j$ to P_c
    end if
end for
P_c : if $\sum_{j=1}^{m} output_j < k$
    $lbound \leftarrow estimate$
    if $ubound - lbound = 1$ then
      return $lbound$
    end if
  elseif $\sum_{j=1}^{m} output_j > k$
    $ubound \leftarrow estimate$
    if $ubound - lbound = 1$ then
      return $ubound$
    end if
  else $\sum_{j=1}^{m} output_j = k$
    return $estimate$
  end if

Table 5. Changed parts of algorithm while performing even times

```
for j = 1 to m
  P_b : compute E_b(a_j) = A
        send A to P_a
P_a : compute E_b(b_j)E_b(a_j) mod n = B
      send B to P_c
P_c : compute
      E_b(b_j)E_b(a_j)E_b(c_j)E_b(mean) mod n = C
      send C to P_b
P_b : decrypt C
        if (f_j + mean) mod n > mean mod n
        return output_j = 0
        send output_j to P_c
      else
        return output_j = 1
        send output_j to P_c
      end if
end for
```

P_b에서 복호화한 값을 P_c로 전송하여 P_c에서 기본 알고리즘과 같이 $output_l$의 값이 k와 일치할 때까지 알고리즘을 수행하며, k와 일치할 때의 $estimate$가 $score$값이 된다.

제안하는 기법에서의 복호화 방식은 기본 기법에서의 복호화 방식과 같다. 그러나 해당 기법에서는 알고리즘이 수행될 때마다 사용되는 공개키가 다르기 때문에 매 라운드마다 복호화하는 사용자가 바뀌게 된다. 따라서 기본 기법에서 P_a에 권한이 집중되었던 것과는 달리 P_a와 P_b가 번갈아가며 복호화를 수행하기 때문에 기본 기법에 비하여 권한이 집중되는 것을 방지할 수 있다.

V. 안전성 분석

본 장에서는 기존의 논문에서 양자간의 덧셈 연산을 다자간의 덧셈 연산으로 확장하면서 Yao의 백만장자 알고리즘을 대신하여 사용된 Paillier 암호화 알고리즘의 안전성에 대하여 알아본다.

기존의 연구에서는 Yao의 양자간의 안전한 비교법을 이용하여 악의적인 사용자와 통신하더라도 틀린 값을 출력할 뿐 본인의 값은 노출하지 않았다.

본 논문에서 제안한 기법 역시 기존의 Yao의 안전한 연산법을 이용한 양자간 연산 알고리즘과 마찬가지로 $score$값이 노출됨으로 인하여 알 수 있는 정보는 상위 k번째 원소가 존재하는 범위와 $(k+1)$번째 원소가 존재할 경우 $(k+1)$번째 원소가 존재하는 범위를 알 수 있을 뿐, 그 이외의 정보는 얻을 수 없다. 또한 악의적인 사용자가 알고리즘에 참여하여 수행할 경우에도 틀린 값을 출력하여 원하는 값을 정확히 알아낼 수 없을 뿐 그 이외의 정보에 대해서는 노출되지 않는다.

또한 알고리즘 수행 시에 새로운 난수를 사용함으로서 공격자에게 t값이 노출되더라도 직접 연산에 사용되는 $estimate$값은 노출되지 않기 때문에 기존의 연구에 비하여 안전성이 떨어지지 않는다. 또한 매 라운드마다 서로 다른 사용자의 공개키를 사용하도록 함으로서, 특정 사용자에게 권한이 집중되는 것을 방지하였다.

기존에 제안된 기법에서는 $score$값과 Yao 비교법을 사용하여 유용한 결과를 도출하였다. 본 논문에서는 Yao 비교법을 대신하여 Paillier 암호화 방식을 사용하여 연산을 수행하였다.

본 논문에서 제안한 기법은 아래의 Table 6.과 같이 연산해야 하는 데이터의 수를 n, 사용자의 수를 m이라고 할 때, 알고리즘 수행횟수는 기존의 양자간의 연구와 동일하다. 그러나 본 논문에서는 기존의 연구를 다자간으로 확장하여 3명 이상의 사용자가 연산을 수행할 경우에도 알고리즘 수행이 가능하도록 하였다. 또한 Paillier 암호화 방식의 사용으로 인하여 기존의 기법에 비하여 연산량은 늘어났으나, 기존의 알고리즘에 비교하였을 경우에도 추가적인 확장이 가능하고, 사용자가 늘어나는 것 외에는 복잡도가 증가하지 않으며, 안전성 면에서도 계산하는 값의 범위 외에는 노출되지 않기 때문에 기존 연구와 거의 유사한 수준의 안전성을 가지게 되었다.

Table 6. Complexity analysis between existing technique and proposed technique(n : number of data)

	[5]	Suggesting Technique
m(number of participants)	$m=2$	$m \geq 2$
performance frequency of algorithm	$O(\log_2 n)$	$O(\log_2 n)$
complexity of algorithm	$O(n^2)$	$O(m \times n^2)$

VI. 결 론

본 논문에서는 기존의 논문에서는 양자 간의 환경에서만 적용하였던 기법을 다자간 환경에서 사용할 수 있도록 확장하였다. 기존의 기법은 A. Yao의 백만장자 문제에 기반 한 알고리즘을 사용하여 사용자가 2명일 경우에만 사용이 가능하였다. 본 논문에서는 Yao 비교법을 대신하여 Paillier 암호화 방식을 적용함으로서, 사용자의 숫자가 늘어나더라도 사용이 가능하도록 개선하였다. 또한 Paillier 암호화 방식을 사용함에 따라 기존의 논문에서 제안하였던 알고리즘에 비하여 뛰어난 안전성을 보장할 수 있게 되었다.

그러나 본 논문에서 제안한 기법은 다자간 환경에서 과반수 이상의 사용자가 공모할 경우 다른 사용자들의 정보가 노출될 수 있는 문제점을 가지고 있다. 따라서 향후 이와 같은 문제를 해결하여 공모에 안전한 기법의 연구가 필요하다.

References

[1] A. C. Yao, "Protocols for secure computations," 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE, pp. 160-164, Nov 1982.

[2] A. C. Yao, "How to Generate and Exchange Secrets," Foundations of Computer Science, pp. 162-167, 27th Annual Symposium on IEEE, pp. 162-167, Oct 1986.

[3] O. Goldreich, Secure Multi-Party Computation, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, June, 1998.

[4] J. Vaidya and C. W. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 639-644, June 2002.

[5] J. Vaidya and C. W. Clifton, "Privacy-Preserving Kth Element Score over Vertically Partitioned Data," Knowledge and Data Engineering, IEEE Transactions on vol. 21, no. 2, pp. 253-258, Feb 2009.

[6] G. Aggarwal, N. Mishra, and B. Pinkas, "Secure Computation of the Kth-Ranked Element," Advances in Cryptology, EUROCRYPT 2004, LNCS 3027, pp. 40-55, 2004.

[7] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Advances in Cryptology, EUROCRYPT 1999, LNCS 1592, pp. 223-238, 1999.

[8] L. Liu, and M. T. Zsu, Encyclopedia of database systems, 2009 Ed., Springer US, pp. 3263-3265, Sep 2009.

[9] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach, "Scalable Semantic Web Data Management Using Vertical Partitioning," Proceedings of the 33rd international conference on Very large data bases, pp. 411-422, Sep. 2007.

[10] S. Goldwasser, "Multi-Party Computations: Past and Present," Proceedings of the 16th annual ACM symposium on Principles of distributed Computing, pp. 1-6, Aug 1997.

[11] M. C. Doganay, and T. B. Pedersen, "Distributed Privacy Preserving k-means Clustering with Additive Secret Sharing," Proceedings of the 2008 international workshop on Privacy and Anonymity in Information Society, pp. 3-11, Mar 2008.

[12] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 86-97, June 2003.

[13] J. W. Byun, A. Karma, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," Advances in Databases: Concepts, Systems and Applications, LNCS 4443, pp. 188-200,

2007.

〔14〕 R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving OLAP," Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 251-262, June 2005.

〔15〕 R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai, "Universally composable two-party and multi-party secure computation," Proceedings of the 34th annual ACM symposium on Theory of computing, pp. 494-503, May 2002.

〔16〕 R. Agrawal, and R. Srikant, "Privacy-preserving data mining," Proceedings of the 2000 ACM SIGMOD international conference on Management of data, vol. 29, no. 2, pp. 439-450, June 2000.

〔17〕 J. Vaidya, and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 206-215, Aug 2003.

〔18〕 W. Diffie, and M. E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. IT-22, no.6, pp. 644-654, Nov 1976.

〔19〕 T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," Advances in Cryptology, LNCS 196, pp. 10-18, Nov 1985.

〔20〕 R. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM vol.21, no.2, pp. 120-126, Feb 1978.

〔21〕 M. O. Rabin, "Digitalized Signatures and Public-Key Functions as Intractable as Factorization," Technical report, MIT/LCS/TR-212, MIT, Jan 1979.

〈저자소개〉

홍 준 희 (Jun Hee Hong) 학생회원
2012년 2월: 고려대학교 정보수학과 졸업
2012년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터마이닝, 비밀 공유 기법

정 재 열 (Jay Yeol Jung) 학생회원
2010년 8월: 고려대학교 정보수학과 졸업
2010년 9월~2013년 8월: 고려대학교 정보보호대학원 석사 졸업
2013년 9월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 비밀 공유 기법

정 익 래 (Ik Rae Jeong) 정회원
1998년 2월: 고려대학교 전산학과 학사 졸업
2000년 2월: 고려대학교 전산학과 석사 졸업
2004년 8월: 고려대학교 정보보호대학원 박사 졸업
2006년 6월~2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~2011년 8월: 고려대학교 정보경영공학전문대학원 조교수
2011년 9월~현재: 고려대학교 정보보호대학원 부교수
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론