The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 15, No. 3, pp.7-13, Jun. 30, 2015. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2015.15.3.7
JIIBC 2015-3-2

DE 알고리즘을 사용한 관계형 데이터베이스를 위한 가역 워터마킹

Reversible Watermarking For Relational Databases using DE (Difference Expansion) Algorithm

김천식*

Cheonshik Kim*

요 약 일반적으로 워터마킹은 오디오, 비디오, 이미지, 그리고 텍스트 등의 콘텐트의 저작권을 위해서 사용되고 있다. 인터넷의 발달로 어플리케이션과 연동되는 관계형 데이터베이스는 악의적인 공격자들에 의한 데이터베이스 복제, 유출 등이 빈번히 발생하고 있다. 따라서 데이터베이스의 저작권보호 역시 필요성이 증대되고 있다. 또한 데이터베이스에는 민감한 개인정보에서 산업기밀 정보까지 다양한 정보고 포함되어 있다. 따라서 관계형 데이터베이스의 보호는 데이터베이스 분야에서 매우 중요한 연구 영역이 되고 있다. 본 논문에서는 관계형 데이터를 보호할 수 있는 다양한 방안의 기존의 연구를 살펴보고, 이들 데이터를 보호할 수 있는 새로운 방법을 제안하고자 한다. 특히 본 논문에서는 가역적인 방법으로 데이터베이스에 워터마크를 삽입하는 방법을 제안하고자 한다. 실험결과 제안한 기술은 악의적인 공격에 강함을 보였다. 또한, 제안한 방법이 실제 어플리케이션에 적용 가능함을 보였다.

Abstract Generally, watermarking can be used copyright for contents such as audios, videos, images, and texts. With the development of Internet, many malicious attackers illegally copy relational databases synchronized applications Therefore, it is needed for the protection of databases copyright, because databases involve various sensitive information such as personal information, information industry, and secret national intelligence. Thus, the protection of relational databases is a major research field in the databases research topics. In this paper, we will review previous researches related the protection of relational databases and propose new method for relational data. Especially, we propose watermarking scheme for databases using reversible method in this paper. As an experimental result, the proposed scheme is very strong to malicious attacks. In addition, we proved our proposed scheme is to apply real application.

Key Words : Watermarking, Copyright, Protection, Databases, DE (Difference Expansion)

Ⅰ. 서 론

소프트웨어, 이미지, 오디오, 그리고 텍스트와 같은 디지털 자산의 무단복제는 이들 자산의 소유자들에게 매우 중요한 관심사이다. 이들 자산의 보호 방법으로 일반적인 것은 주로 디지털 워터마크를 데이터에 삽입하는 것이다[1-6]. 워터마킹 소프트웨어는 보호할 객체의 값을 변경하여 워터마크를 삽입하는 기술이다.

*종신회원, 안양대학교 디지털미디어학과
접수일: 2015년 4월 23일, 수정완료일: 2015년 5월 23일
게재확정일: 2015년 6월 12일

Received: 23 April, 2015 / Revised: 23 May, 2015 /
Accepted: 12 June, 2015
*Corresponding Author: mipsan@paran.com
Dept. of Digital Media Engineering, Anyang Univ., Korea

워터마크는 데이터의 사용에 중대한 영향을 주어서는 안 된다. 또한, 워터마크를 제거하려는 공격자에게 도출되지 말아야한다. 응용프로그램의 일부로 사용되는 데이터베이스의 사용빈도가 높아짐에 따라서 워터마킹이 포함된 데이터베이스의 필요성이 증대되고 있다[7-8].

인터넷의 활성화에 따라서 많은 데이터는 원격지에서 접속되고 검색될 수 있도록 가능하게 되었다. 이와 같은 트렌드는 최종사용자에게 유용한 반면, 외부의 많은 공격자에게 이러한 데이터의 노출에 대한 위험이 증가하게 되었다. 이런 이유로, 데이터가 원본데이터임을 증명하는 식별기능이 요구되고 있다.

다양한 데이터마이닝에 사용되는 데이터 값, 기상데이터들에 워터마킹을 삽입함으로써 원본 데이터에 대한 식별기능 및 소유권 확인이 가능하다. 이러한 데이터들은 결과에 큰 영향을 주지 않기 때문에 워터마킹에 적합한 데이터이다.

본 논문에서 우리는 데이터베이스 분야의 연구영역으로 의미가 있는 워터마킹을 이용한 관계형 데이터베이스의 데이터의 저작권관리 방법을 제안하고자 한다.

II. 관련 연구

1. 데이터베이스 워터마크 방법들

데이터베이스를 위한 워터마크의 시작은 Agrawal과 Kiernan[7]에 의해서 시작되었고 강력한 워터마크 방법을 제안했다. 이들이 제안한 방법은 주로 관계형 데이터의 숫자 속성을 위한 워터마킹 방법에 집중했다. 따라서 이들 숫자 속성들이 적은 량의 데이터 수정을 견딜 수 있다는 전제하에 가능한 방법이다. 첫 레코드와 레코드내의 속성은 워터마크 은닉을 위해 무작위로 선택되며 선택된 값들은 워터마크 삽입을 위해 수정이 필요하다. Radu Sion[8] 등은 데이터베이스 워터마킹 분야에 중요한 기여를 했다. 즉, 숫자 데이터와 분류 데이터에 워터마크를 삽입하는 방법을 제안했다.

이 방법에 따르면, 숫자 데이터는 비밀 키를 이용해 서부분집합으로 나누어 1비트 데이터를 각 부분 집합에 은닉한다. Radu Sion은 자신이 제안한 방법이 다양한 데이터베이스 워터마킹 공격 (서브셋 공격, 데이터 재정렬, 변환공격)에 강함을 주장했다. 반면에 Radu Sion이 제안한 방법은 빈번한 갱신을 요구하는 시스템 응용에는 효율성이 떨어지는 문제점이 있다. Yingjiu Li[9] 등은 관계형 데이터 인증을 위한 섬세한(fragile) 워터마킹 방법을 제안했다.

Shehab 등[10]은 숫자 데이터를 위한 최적화 워터마크를 제안했다. 최적화를 위해서 유전자알고리즘과 패턴 탐색을 활용했다. 이 기술은 데이터에 최소 왜곡과 기존의 방법보다 더 강한 공격에 대응 가능한 최신의 기술이다. 이 방법의 단점은 숫자 데이터로 제한된다는 것이다. Sion[8]은 분류 데이터 보호를 위해서 강력한 워터마킹 방법을 제안했다.

관계형 데이터를 위한 워터마킹도 역시 이미지에 적용될 때와 비슷한 가능이 필요하다. 즉, 이미지의 각 픽셀의 값들이 변경되더라도 이미지의 질이 떨어지지 않는 것과 같이 데이터베이스의 숫자데이터의 값이 변경되는 것이 질의(Query) 결과에 큰 영향을 주지 않는다는 가정이 필요하다. 그러므로 이 변경은 데이터의 사용에 영향을 주지 않아야한다.

2. Agrawal과 Kiernan의 워터마킹 방법[7]

기호설명
K: 비밀키
e: 사용에 영향을 받지 않는 변경 가능한 LSB의 수, 예) e=3, 101101101.1011101
m: 마킹(marking)의 개수로 마커선택을 위해 사용됨
v: 워터마킹 과정에서 사용되는 속성의 개수

일 방향 해시함수 H는 임의의 길이의 메시지 M에 대해서 고정길이의 값 h를 반환한다. 이 함수의 특징은 다음과 같다.

- 주어진 M에 대해, h를 계산하는 것이 간단하다.
- 주어진 h에 대해, M을 계산하는 것이 어렵다.
- 주어진 M에 대해, $H(M)=H(M')$인 경우에 메시지 M'을 알아내는 것은 어렵다.

메시지 인식 코드(MAC)은 키 값에 의존하는 단방향 함수이다. 즉, 다음과 같은 계산이 성립한다.

$$MAC(r.P)=MAC(r.P)=H(K\|MAC(K\|r.P) \quad (1)$$

$r.P$는 관계형 릴레이션의 기본 키 속성이고, K는 소

유자에게만 알려진 비밀 키이고, 함수의 결과 값은 정수 값이다.

```
(1) 워터마크 삽입 알고리즘
For all tuples r in D {
  MAC(r.P) = MAC(r.P) = H(K || MAC(K||r.P)
  if(MAC(r.P) mod m == 0) {   // Marker Selection
     i = (MAC(r.P) mod v       // Selected Attribute
     b = (MAC(r.P) mod e       // Selected LSB index
     if((MAC(r.P) mod 2 == 0)  // MAC is even
        Set bit b of r.Ai
     Else
        Clear bit b of r.Ai
   } // end if ://~
} // end for ://~
```

워터마크 삽입을 위해 $MAC(r.P)$를 계산한다(수식 1). $MAC(r.P)$는 기본 키에 대해서 단방향 해시 값을 구하는 함수이다. "$MAC(r.P) \bmod m$"가 0과 같으면 그림 1의 릴레이션에서 속성위치 i를 계산하고 LSB 인덱스 b를 계산한다. $MAC(r.P)$의 값이 짝수이면 i번째 속성의 레코드 값에 비트 b를 설정한다.

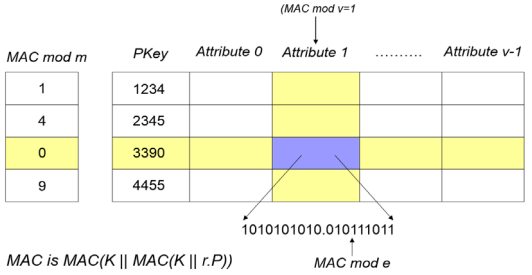


그림 1. (1) 워터마크 삽입 알고리즘 예제
Fig. 1. (1) Example of watermark insert algorithm

```
(2) 워터마크 탐지 알고리즘
Match = Total_Count = 0
For all tuples r in D {
 r.MAC = H(K||r.P||K)
 if(r.MAC mod m == 0) // Marker Selection
     Total_Count++
     i = r.MAC mod v    // Selected Attribute
     b = r.MAC mod e    // Selected LSB index
     if(r.MAC mod 2 == 0)  // MAC is even
        if (bit b of r.Ai is Set)  Match++
        Else If (bit b of r.Ai is Clear)  Match++
}
Compare (Match/Total_count) > Threshold
```

워터마크 탐지과정의 첫 번째 단계로 $Match = Total_Count = 0$로 설정한다. 다음으로 레코드의 개수만큼 반복해서 $r.MAC$을 구하다. $(r.MAC \bmod m)$=0이면 $Total\_Count$++를 한다. 워터마크 삽입과정과 유사하게 i, b를 계산하고 $r.MAC$의 값이 짝수 및 홀수에 따라 알고리즘과 같이 처리하면 된다.

이 방법의 강점은 계산시간이 $O(n)$으로서 레코드가 정렬될 필요는 없다. 단점은 데이터 공격자가 LSB 값을 변경하면 간단히 워터마크가 제거된다. 또한, 기본 키가 반드시 필요하다.

3. Al-Haj와 Odeh의 워터마크 방법[11]

제안한 알고리즘은 워터마크 이미지를 관계형 데이터베이스에 은닉하는 방법으로 릴레이션의 각 레코드가 워드의 집합으로 이루어진 경우에 적합한 방법이다. 이 방법은 이진 이미지를 일정한 크기로 나눈 다음 나누어진 이진 비트들을 10진수로 변경하고 10진수를 단어들 사이에 공백의 크기를 이용하여 표기하는 방법으로 워터마크를 삽입하는 방법이다. 공백을 이용한 워터마크의 장점은 큰 용량의 워터마크도 삽입이 가능하다는 점이다. 또한 이 방법은 큰 용량의 워터마크를 릴레이션의 여러 곳에 분산하여 삽입하는 것도 가능하다. 이와 같은 장점 때문에 워터마크가 쉽게 제거되거나 파괴되지 않을 수 있다. 제안한 알고리즘은 두 과정으로 워터마크 삽입과 워터마크 추출의 과정이다. 삽입과정의 설명은 다음과 같다.

워터마크 삽입과정

[단계 1]: 워터마크 이미지를 n개의 길이를 갖는 m 문자로 만든다.

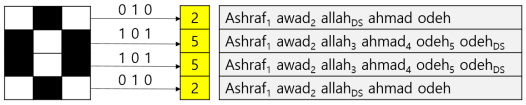
[단계 2]: 데이터베이스를 논리적으로 레코드의 부분집합으로 나눈다. 부분집합은 m개의 레코드로 구성된다.

[단계 3]: 레코드 i번째에 은닉할 m_j의 문자열의 10진수에 해당하는 속성 A_k번째의 단어에 이중 공백(DS: double space)를 넣는다(그림 2참고).

[단계 4]: 단계 3을 반복 적용한다. 더 이상 삽입할 m_j가 없으면 삽입과정을 종료한다.

그림 2는 이진 4x3 이진 워터마크를 데이터베이스

레코드에 은닉하는 방법을 간단히 설명한 것이다. 이진 이미지의 각 행에 3비트 값을 10진수로 변환 한 것이 그림 2.(a)이고 2.(b)는 레코드의 단어에 이진 이미지를 은닉한 것을 보인 것이다. 그림 2.(b)에서 DS전에 공백의 개수를 의미하는 10진 숫자는 이미지의 각 행의 10진 수자와 일치한다.



(a) 4x3 이미지 (b) 텍스트에 DS로 10진수 삽입

그림 2. (a) 이진 이미지 워터마크와 이것의 10진수, (b) 워터마크 예제로 첨자는 공백의 개수이다.

Fig. 2. (a) Binary image watermark and its' decimal number, (b) Example of watermarked words with spaces and suffix is number of space.

그림 3은 워터마크 은닉과정을 완료한 후 관계형 데이터베이스의 모습을 그림으로 나타낸 것이다. 그림 3에서 각 레코드는 속성 A들의 집합으로 구성되며 속성의 도메인은 단어들의 집합으로 구성된다.

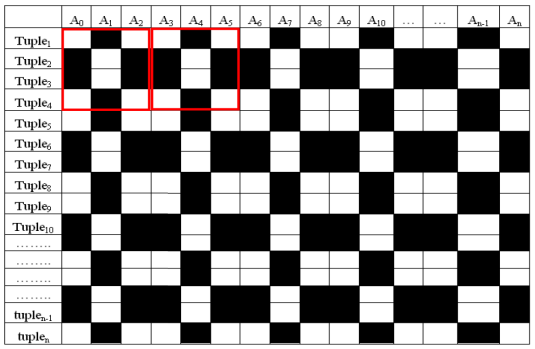


그림 3. 워터마크가 삽입된 데이터베이스의 가상화 이미지

Fig. 3. Virtual image of a database with watermark.

Ⅲ. 제안 방법

본 논문에서 제안한 방법은 기존에 이미지에서 워터마크를 삽입하던 방법으로 원본 이미지를 복원하는 기능이 있는 가역적인 데이터은닉 기법인 DE (Difference Expansion)[12]을 데이터베이스 릴레이션에 적용하고자 한다.

본 논문에서 제안하는 방법이 성립하기위한 전제는 그림 4와 같은 릴레이션으로 속성1부터 속성n까지로 구성되며 각 도메인은 정수형 데이터로 값의 수정이 1비트인 경우 질의 및 데이터베이스 성능에 영향을 주지 않는 경우로 한정한다.

기호설명
x: 홀수 속성필드의 숫자 값
y: 짝수 속성필드의 숫자 값
l: x와 y의 평균으로 즉, $\lfloor (x+y)/2 \rfloor$
$\lfloor . \rfloor$ 는 floor 함수
h: x와 y의 값의 차이
b: 삽입할 워터마크 비트

예제 1: 워터마크 삽입하는 과정을 예제로 설명하고자 한다.

기본 키가 1001인 레코드의 속성1과 속성2는 각각 x와 y로 배정되며 $x=206$, $y=201$이라고 가정하며 은닉할 워터마크 비트는 $b=1$이다. 먼저 평균 l과 차이 값 h를 계산한다.

$$l=\left\lfloor \frac{206+201}{2} \right\rfloor = \left\lfloor \frac{407}{2} \right\rfloor = 203, h=206-201=5$$

h를 이진수로 나타내면 $h=5=101_2$이 된다. 이후 b를 h의 LSB의 다음에 추가한다. 그럼 h'는 수식으로 다음같이 나타낼 수 있다.

$$h' = 101b_2 = 1011_2 = 11.$$

다음으로 원본 (x, y)에 대해서 데이터가 은닉된 새로운 (x', y')를 다음의 과정으로 계산가능하다.

$$x' = 203 + \left\lfloor \frac{11+1}{2} \right\rfloor = 209, y' = 203 - \left\lfloor \frac{11}{2} \right\rfloor = 198$$

데이터가 은닉된 새로운 속성1의 $x'=209$와 새로운 속성2의 $y'=198$가 계산을 통해서 완성되었다. (x', y')는 워터마크가 삽입된 속성이다.

Primary Key	속성 1	속성 2	...	속성 n
1001	x	y		
1002	x	y		
....	x	y		
9999	x	y		

그림 4. 워터마크 삽입을 위한 관계형 데이터베이스 릴레이션

Fig. 4. Relational database for watermark insertion.

예제 2: 워터마크를 추출하는 과정을 예제로 설명하고자 한다.

그림 4의 기본 키 1001 레코드에 대한 속성1과 속성

2로부터 은닉된 워터마크 비트를 추출하는 과정을 알아보자. 속성1과 속성2의 값 (x', y')로부터 은닉된 비트를 추출하고 원래의 (x, y)로 복원하기 위한과정으로 먼저 평균과 두 값의 차이 값을 알아낸다.

$$l' = \left\lfloor \frac{209+198}{2} \right\rfloor = 203, h' = 209-198 = 11.$$

h'에 대해서 이진수로 변환한다. 즉, $h' = 11 = 1011_2$.

h'의 LSB는 은닉된 1비트의 데이터이고 수식으로 다음과 같이 나타낼 수 있다.

$$b = LSB(h') = 1, h = \left\lfloor \frac{h'}{2} \right\rfloor = 5.$$

이전에 계산된 l'와 h를 이용하여 기본 키 1001에 해당하는 원본 릴레이션 속성1, 속성2의 레코드를 복원할 수 있다.

$$x = 203 + \left\lfloor \frac{5+1}{2} \right\rfloor = 206, y = 203 - \left\lfloor \frac{5}{2} \right\rfloor = 201$$

릴레이션의 원본속성 (x, y)가 위의 과정으로 복원되었다. 지금까지 워터마크 은닉과 추출과 관련한 과정을 예를 통해서 보였다. 이의 과정을 위한 일반화 수식은 수식 (1)(2)와 같다.

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, h = x-y \quad (1)$$

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2)$$

워터마크 은닉과 복원의 과정에서 사용된 정수변환은 (1)(2)는 정수 Haar 웨이블릿 변환 또는 S 변환이라 부른다.

Ⅳ. 실험 및 결과

본 논문에서는 제안한 방법을 실험하기 위해서 릴레이션을 구축한 후에 워터마크를 삽입하였다. 실험에 사용한 환경은 윈도우즈 7과 Visual C++이다. 구축한 릴레이션에 워터마크를 삽입하고 삭제하는 과정에 대한 평가를 위해서 워터마크가 삽입된 릴레이션을 무작위로 삭제한 후 워터마크의 상태를 측정하는 것으로 하였다.

이 실험에서 무작위로 선택한 레코드들이 삭제한 후에 워터마크의 원본과 비교한다. 릴레이션에서 레코드를 2% ~ 30%까지 삭제하면서 삭제비율에 해당하는 워터마크의 왜곡정도를 차트로 나타낸 것이 그림 5의 모습이다. 그림 5에서 x좌표는 레코드 삭제비율이고 y좌표는 워터마크 왜곡비율이다. 25%의 릴레이션이 삭제되었을 때 워터마크는 12%까지 왜곡이 있었다.

Mallory의 워터마크에 대한 공격의 우선순위는 워터마크의 파괴에 있다. 공격자가 만일 릴레이션에 대한 사전지식이 없다면 공격자는 워터마크를 제거하겠다는 생각으로 릴레이션의 임의의 장소를 수정하는 것으로 목적을 이루려고 할 것이다.

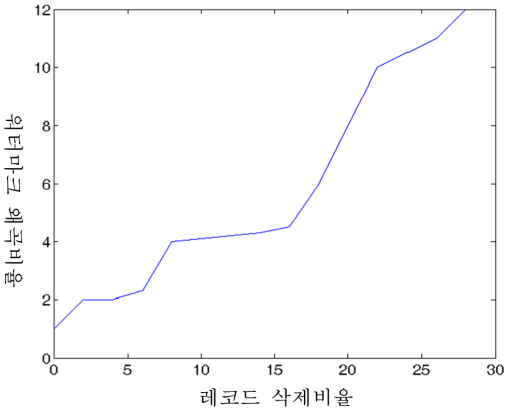


그림 5. 레코드 삭제공격에 대한 워터마크 왜곡상태
Fig. 5. The state of distortion from the delete attack

이 실험에서 우리는 워터마크가 포함된 릴레이션에 대한 무작위 갱신공격에 대한 워터마크의 손실을 분석한다.

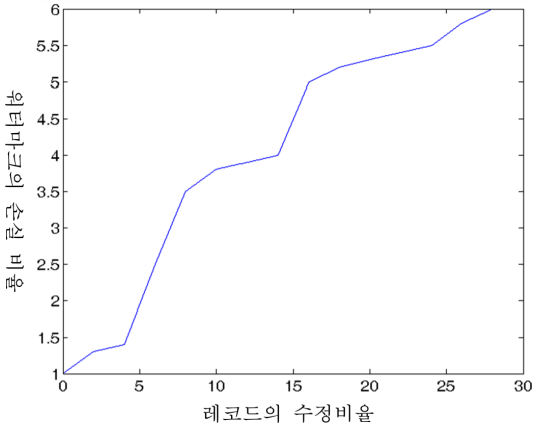


그림 6. 레코드 수정공격에 대한 워터마크의 손실 비율
Fig. 6. The loss rate of watermark from the record modifying attack

그림 6은 제안한 실험방법에 따라 실험한 결과를 차트로 나타낸 것이다. x좌표는 **레코드의 수정 비율**이고 y좌표는 **워터마크의 손실 비율**을 나타낸다. 실험결과 27%의 수정에 대해서 6%의 워터마크 왜곡이 있었다.

워터마크를 삽입하기 전에 기본 키에 대한 해시 값을 구한 후 이를 이용하여 릴레이션의 레코드를 정렬한다. 반대로 워터마크를 추출할 때도 역시 이와 같은 과정을 거치기 때문에 정렬 공격에 강한 장점이 있다.

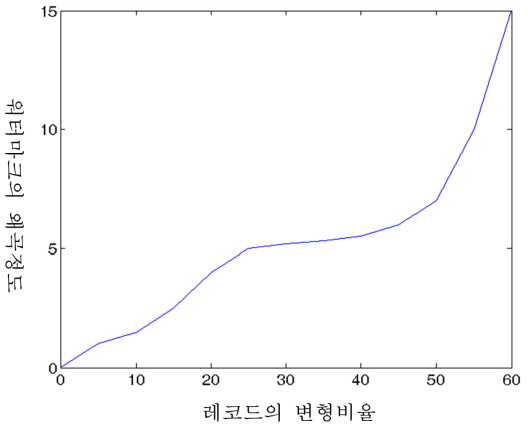


그림 7. 선택공격
Fig. 7. Selection attack

실험에서 우리는 무작위로 선택한 속성의 값을 변경한 후 워터마크를 추출하고 은닉한 원본 워터마크와 비교한다. 그 결과가 그림 7과 같다. 제안한 방법이 이러한 종류의 공격에도 강한 면을 보여주고 있다. 레코드의 60%를 바꾸었을 때, 워터마크의 변형이 15%이하의 변형을 보였다. 그림 7에서 x축은 레코드의 변형비율이고 y축은 워터마크의 왜곡정도를 퍼센트로 나타낸 것이다.

V. 결 론

본 논문에서 우리는 관계형 데이터베이스의 릴레이션에 대한 소유권을 보호하기 위한 방법으로 릴레이션에 워터마크를 삽입하는 방법으로 소유권을 확인할 수 있도록 하였다. 제안한 방법은 다음과 같다.

새로운 워터마크방법을 제안함.

워터마크에 대한 공격에 강한 대응이 가능함.

우리는 관계형 데이터베이스 릴레이션의 저작권을 보호할 수 있는 효율적인 워터마킹 기술을 제공하였다. 실험을 통해서 우리는 제안한 방법이 일반적인 데이터베이스 공격방법에 강함을 보였다. 향후계획은 섬세한(fragile) 워터마크를 사용하여 데이터 인증을 위한 방법을 개발하는 것이다.

References

[1] Cheonshik Kim, "Reversible Data Hiding based on QR Code for Binary Image," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), vol.14, no. 6, pp.281-288, 2014.

[2] Cheonshik Kim, "Data Hiding Based on BTC using EMD," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), vol.14, no.2, pp.11-16, 2014.

[3] B.H. Lee, "Technique for production and encoding of New dot-type Print Watermark Pattern," Journal of the Korea Academia Industrial cooperation Society, vol.10, no.5, pp.979-984, 2009.

[4] Cheonshik Kim, Ching-Nung Yang, "Watermark with DSA signature using predictive coding," Multimedia Tools and Applications, DOI:10.1007/s11042-013-1667-6, pp.1-15, 2013.

[5] Cheonshik Kim, Dongkyoo Shin, Dongil Shin, Xinpeng Zhang, "Secure and Trust Computing, Data Management and Applications," Communications in Computer and Information Science Volume 186, pp.130-138, 2011. 2011.

[6] Cheonshik Kim, "Data Hiding Based on Compressed Dithering Images," Advances in Intelligent Information and Database Systems Studies in Computational Intelligence Volume 283, pp 89-98, 2010.

[7] R. Agrawal and J. Kiernan. Watermark relational databases. In Proc. of the 28th International. Conference. On Very Large Data

Bases, 2002.
[8] R.Sion, M. Atallah, and S. Prabhakar. Rights protection for relational data. In Proceedings of ACM SIGMOD 2003, 2003.
[9] Y. Li, H. Guo, S. Jajodia, Tamper detection and localization for categorical data using fragile Watermarks in: 4th ACM Workshop on Digital Rights Management, CCS04, October 2004.
[10] Mohammad Shehab, Elisa Bertino, Arif Ghafoor Watermarking Relational Data using Optimization Based Techniques, IEEE Transactions on Knowledge and Data Engineering, vol.20, no.1, pp.116-129
[11] Ashraf Odeh, Ali Al-Haj, "Watermarking Relational Database Systems," IEEE, pp. 270-274, 2008.
[12] Jun Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.8, pp.890-896, 2003.

저자 소개

김 천 식(종신회원)

- 1997년 : 한국외국어대학교 컴퓨터 및 정보통신공학과 (공학석사)
- 2003년 : 한국외국어대학교 컴퓨터 및 정보통신공학과 (공학박사)
- 2010년 ~ 2012년 : 세종대학교 교수
- 2013년 ~ 현재 : 안양대학교 교수
- 2007년 ~ 2009년 : 대한전자공학회 컴퓨터소사이어티 멀티미디어 분과위원장
- 2012년 : UMAS 워크샵 프로그램 의장
- 2013년 : GPC 2013 프로그램 의장
- 2014년 : FutureTech 2014 프로그램 의장
- 2015년 : IoP2015 TPC 위원
- 2015년 : 3PGCIC 2015 세션 의장
- 2012년 ~ 현재 : TACT 영문 저널편집 위원
- 2015년 : 대한전자공학회 컴퓨터분과 협동부회장

<주관심분야 : 데이터베이스, 데이터마이닝, Steganography, 영상처리, e-Learning>