ElGamal함수를 사용하는 디지털 이미지 워터마킹 기법

이 진 호[†]·김 태 윤[††]

요 약

디지털 이미지 워터마킹(digital image watermarking)은 이미지 소유자의 정보를 디지털 이미지 속에 삽입시켜 이미지 소유자의 저작권을 보호하는 것을 목적으로 하는 기법이다. 저작권 보호를 위한 디지털 이미지 워터마킹 기법은 기존의 스테가노그라피(steganography)보다 워터마킹 공격에 대한 견고성과 육안적 비구별성을 동시에 추구해야 하고, 워터마킹 알고리즘의 은닉성 대신 키의 은닉성이 보장되어야 하며, 암호학과 마찬가지로 키의 사용으로 허가받지 않은 사용자의 워터마크 검출을 방지할 수 있어야 한다. 본 논문에서는 암호학 함수인 ElGamal함수를 사용하는 워터마킹 기법을 제안한다. 일방향 해쉬 함수를 구현하기 위해 ElGamal일방향 함수와 모듈라 연산을 사용한다. 제안하는 워터마킹 기법은 LSB(least significant bit)공격과 감마 보정 공격에 대해 견고하며 육안적 비구별성(perceptual invisibility)이 높다. 제안하는 워터마킹 기법의 실제 구현 및 실험을 통한 실험 결과를 분석하여 견고성과 육안적 비구별성의 특징을 확인한다. 향후 과제로, 키생성을 위한 의사난수성과 비대칭키의 생성을 동시에 달성시키는 알고리즘 연구가 요구된다.

A Digital Image Watermarking Scheme using ElGamal Function

Jean-Ho Lee[†] · Tai-Yun Kim[††]

ABSTRACT

Digital image watermarking is a technique for the purpose of protecting the ownership of the image by embedding proprietary watermarks in a digital image. It is required for the digital image watermarking scheme to pursue the robustness against water marking attacks and the perceptual invisibility more than usual in steganography area, to guarantee not a hidden watermarking algorithm but the publicity of watermarking algorithm details and hidden use of key, which can protect the unauthorized user access from detection. In this paper we propose a new copyright watermarking scheme, which is based on one-way hash functions using ElGamal functions and modular operations. ElGamal functions are widely used in cryptographic systems. Our watermarking scheme is robust against LSB(least significant bit) attacks and gamma correction attack, and also perceptually invisible. We demonstrate the characteristics of our proposed watermarking scheme through experiments. It is necessary to proceed as the future work the algorithm of achieving at the same time both the pseudo-randomness for the stego-key generation and the asymmetric-key generation.

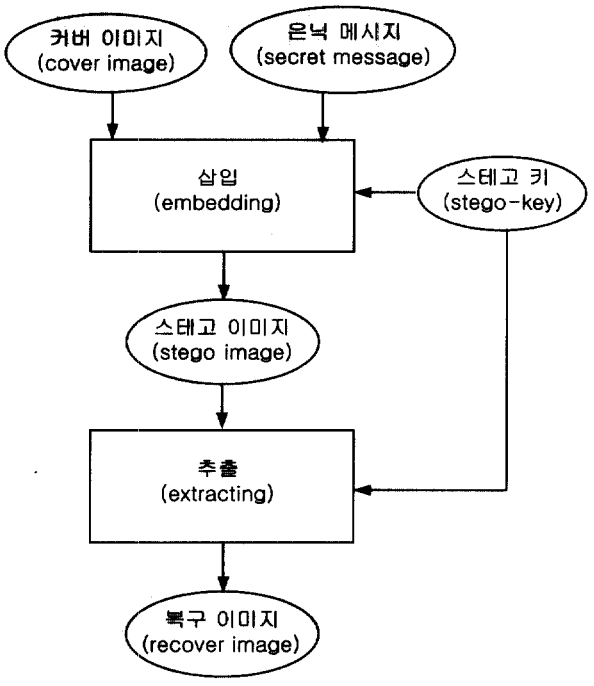
키워드 : 스테가노그라피(Steganography), 디지털 워터마킹(Digital Watermarking), 일방향 해쉬 함수(One-way Hash function), 엘가말 함수(ElGamal function)

1. 서 론

디지털 워터마킹은 디지털 이미지 데이터의 저작권을 보호하고 불법 접근을 방지하는 것을 목적으로 고안되었다[7]. 디지털 워터마킹 기법은 커버(cover)라고 하는 디지털 데이터 상에 키(stego-key)를 사용하여 워터마크(watermark)라고 하는 비밀 정보(secret embedded message)를 삽입하는 기법으로 키를 가진 합법적인 사용자만이 삽입했던 비밀 정보를 추출해 낼 수 있다. 비밀 정보 비트가 삽입되는 커버 비트를 은닉 비트(hiding-bit)라고 하고, 워터마크가 삽입된 커버 데이터를 스테고 데이터(stego data)라고 한다. 커버 데이터는 이미지, 오디오, 동영상, 텍스트 등 모든 멀티미디어 데이터를 포괄하며 비밀 정보는 주로 저작권을 나타낼 수 있는 로고나 문구가 사용된다. 디지털 이미지 워터마킹은 커버 데이터로 이미지 데이터를 사용한다. 디지털 이미지 워터마킹 기법의 전체적인 처리 과정을 도식화하면 (그림 1)과 같다.

디지털 이미지 워터마킹 기법은 기존의 스테가노그라피(steganography) 기법과 동일하게 커버 이미지 데이터에 비밀 정보의 삽입과 추출 과정을 거쳐 동작한다. 스테가노그라피의 목적은 스테고 데이터 안에 은닉 정보를 삽입하여 은밀한 교신을 하기 위한 것이고, 디지털 이미지 워터마킹은 은닉 정보에 커버 이미지 데이터의 저작권을 표시할 수 있는 데이터를 포함하여 디지털 이미지 데이터의 소유권을 판별하는 것을 목적으로 한다. 디지털 이미지 워터마

† 준 회 원 : 고려대학교 대학원 컴퓨터학과
†† 종신회원 : 고려대학교 컴퓨터학과 교수
논문접수 : 2001년 6월 1일, 심사완료 : 2001년 10월 11일



(그림 1) 디지털 이미지 워터마킹 처리 과정

킹 기법은 추가적인 특성이 요구되는데, 스테고 이미지를 손상시킴으로써 스테고 이미지에 포함된 저작권을 표시하는 은닉 정보를 손상시키려는 공격에 대한 견고성(robustness)을 보장해야 하고, 은닉 정보에 대한 접근을 신분이 보장된 사용자로 제한하고 불법적인 사용을 방지할 수 있어야 한다.

디지털 이미지 워터마킹 기법은 정보를 은닉한다는 점에서 암호학과 목적이 동일하지만 암호학과는 달리 스테고 데이터의 형태가 변하지 않는다는 점이 다르다. 암호학에서는 커버 데이터에 비밀 정보를 삽입시킨 스테고 데이터는 원래의 커버 데이터와 전혀 다른 구조와 형태를 갖게 된다. 디지털 이미지 워터마킹 기법에서는 커버 데이터의 손상 정도가 육안으로 구별되지 않아야 하기 때문에, 커버 데이터와 스테고 데이터 사이의 차이가 육안적으로 구별이 어려워야 하는 특성이 존재한다. 스테고 데이터에서 은닉 정보를 추출할 때, 삽입시 사용했던 커버 이미지를 사용하지 않고 스테고 키만을 사용해서 추출할 수 있어야 한다.

이미지를 스테고 데이터로 사용하는 경우 일반적인 이미지 처리 과정에서 생길 수 있는 이미지 화질의 저하로 인한 스테고 이미지 데이터의 손상과 같은 비의도적인 공격 이외에도, 스테고 이미지 데이터를 손상시켜 스테고 이미지 안에 포함된 은닉 메시지 자체를 파괴시키려는 의도적인 공격이 시도될 수 있다. 디지털 이미지 워터마킹 기법은 스테고 이미지 데이터에 대한 비의도적인 공격과 의도적인 공격에 대해 은닉 데이터가 파괴되지 않도록 견고성을 보장할 수 있어야 한다.

디지털 이미지 워터마킹 기법에서 은닉 정보의 위치를 결정할 때 다양한 공격에 대한 견고성과 육안적 비구별성을 동시에 만족시켜야 하지만 2가지 성질 사이에는 모순성(trade-off)이 존재한다. 견고성을 높이기 위해 은닉 정보를 중복해서 삽입시키면 커버 이미지 화질의 저하를 가져오고, 육안적 비구별성을 높이기 위해 은닉 정보의 양을 가능한 최소로 하여 커버 이미지 내의 중요 지역에만 집중적으로 삽입시켜 지역성(locality)을 증가시키면 워터마킹 공격에 대한 견고성이 낮아지게 된다.

디지털 이미지 워터마킹 기법은 이미지 정보의 처리 방식에 따라, 공간 영역 처리와 주파수 영역 처리로 나누어 볼 수 있다. 공간 영역 방식에서 커버 이미지 안의 은닉 정보 삽입 위치의 지역성을 낮게 유지하려면 은닉 정보의 삽입 위치가 의사난수성(pseudo-randomness)을 만족시켜야 한다[1]. 커버 데이터 비트의 인덱스 원소로 구성된 의사 난수 순열(pseudo-random permutation)을 형성하고, 이 순열 집합을 가지고 커버 이미지 안에서 은닉 데이터 비트의 위치를 결정하게 된다.

본 논문에서는 의사 난수 순열 집합의 선택 기법에 기반하여 은닉 비트 위치를 결정하는 디지털 이미지 워터마킹 기법을 제안한다. 본 논문에서는 의사 난수 순열을 생성하기 위해 일방향 해쉬 함수를 의사 난수 생성기로 사용하고, 일방향 해쉬 함수를 구현하기 위해 일방향 함수와 모듈라 연산을 함께 사용한다. Merkle의 주장[10]에 따라 암호화 함수가 일방향 해쉬 함수를 구현할 수 있다는 연구 결과에 기초하여 ElGamal 함수를 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 디지털 이미지 워터마킹 기법들의 특징과 ElGamal 함수에 대해 살펴본다. 3장에서 새로운 디지털 이미지 워터마킹 기법을 제안하고, 4장에서는 실험을 통해 제안한 알고리즘의 성능과 안전성을 분석한다. 5장에서는 결론 및 향후 과제를 제시한다.

2. 관련 연구

2.1 디지털 이미지 워터마킹 기법

디지털 이미지 워터마킹 기법은 이미지 처리 영역에 따라 공간 영역(spatial domain) 방식과 주파수 영역(frequency domain) 방식으로 나눌 수 있다. 공간 영역 방식에서는 이미지의 영상 정보를 이미지 상의 평면 위치 정보를 기준으로 저장되기 때문에 색상 정보를 접근하려면 원하는 구역의 평면상의 위치 정보가 먼저 정의되어야 한다. 공간 영역 방식에서 이미지 처리 알고리즘은 간단하고 구현이 쉬운 장점이 있지만, 커버 이미지의 크기가 은닉 정보 비트들을 삽입시킬 수 있는 범위로 제한되기 때문에 커버 이미지 데이터에 삽입시킬 수 있는 은닉 정보의 양에 한계가 있다. 공간 영역에서 사용되는 대표적인 디지털 이미지 워터마킹 기법으로 픽셀(pixel) 데이터의 LSB(least significant bit)들

만을 은닉 정보 삽입에 이용하는 LSB기법이 있다. LSB기법은 구현이 쉽고 육안적 비구별성(perceptual invisibility)이 높은 반면 견고성(robustness)이 낮고 은닉 이미지 정보 크기의 제약에 대한 단점이 있다.

주파수 영역 방식은 이미지 정보를 이미지의 공간적 위치가 아니라 주파수 영역으로 변환하여 정의하는 방식으로 샘플링(sampling)과 양자화(quantization) 과정을 거쳐 얻은 계수(coefficient)값을 연산하여 저장한다. 주파수 영역 변환 기법은 DCT(discrete cosine transform), DFT(discrete fourier transform), DWT(discrete wavelet transform)과 같은 변환 함수를 사용한다. 공간 영역 방식의 문제점인 이미지 처리 과정에서 발생할 수 있는 워터마킹 공격(dithering, smoothing, converting, compression, filtering등)에 대해 주파수 영역 방식에서는 방어할 수 있는 알고리즘 개발이 다양하고 쉬워 견고성이 높고 은닉 정보의 크기에 대한 제약이 없다는 장점이 있지만 구현이 어렵다는 단점이 있다.

스테고 이미지의 화질을 높이기 위해 육안적 비구별성을 증대시키는 방법으로 인지 마스킹(perceptual masking) 기법이 도입되었다. 인지 마스킹 기법에서는 커버 이미지 영역에서 육안으로 구별가능한 영역을 결정하여 그 부분에 은닉 정보를 삽입하고 나서 스테고 이미지와 커버 이미지의 동일한 부분을 비교하여 차이가 발생하는 위치의 색상값의 평균을 취한다. 인지 마스킹 기법은 커버 이미지와 스테고 이미지 사이의 화질 손상 차이를 줄이기 때문에 육안적 구별성이 낮아진다. 인지 마스킹 기법은 피드백(feedback) 과정이기 때문에 독립적인 워터마킹 기법으로 사용되기는 어렵고 공간 영역 방식이나 주파수 영역 방식과 함께 혼합된 방식으로 사용된다.

2.2 ElGamal 함수

ElGamal 함수는 암호학적인 기본 요소로서 암호화(encryption) 기법과 복호화(decryption) 기법이 존재한다[14]. 먼저 1개의 솟수 p와 2개의 임의의 수 g, k를 선택하며, 이때 $g,k < p$를 만족한다. (k, g, p)를 이용하여 $y = g^k \bmod p$를 계산한다. 이렇게 얻은 (y, g, p)는 공개키가 되고, k가 개인키가 된다.

평문 메시지를 M, M을 암호화한 암호문을 C라고 표시한다.

- 암호화 기법 : 먼저 (p-1)과 서로소(relatively prime) 관계인 임의의 수 r을 선택한다. 메시지 M을 암호문 C로 암호화시키기 위해 임의의 수 r과 공개키를 사용한다.

  $a \equiv g^r \bmod p$

  $b \equiv y^r M \bmod p$

  이렇게 얻은 (a,b)가 메시지 M의 암호문 C가 된다.
- 복호화 기법 : 암호문 C를 복호화시키기 위해 $M' \equiv b/a^k \bmod p$를 계산한다.

$M' \equiv b/a^k \bmod p \equiv y^r M/a^k \bmod p \equiv g^{kr} M/g^{kr} \bmod p \equiv M \pmod p$

본 논문에서 제안하는 알고리즘은 ElGamal함수를 가지고 계산한 값들을 모듈라 연산을 시켜서 임의 수(random number)를 생성시킨다.

3. 제안한 워터마킹 기법

본 장에서는 일방향 해쉬 함수를 사용하는 새로운 워터마킹 기법을 제안한다. 커버 데이터의 은닉 비트 집합을 생성하기 위해 ElGamal함수와 모듈라 연산을 사용하여 일방향 해쉬 함수를 구현한다. 의사 난수 순열을 얻기 위해 ElGamal함수를 사용하고, ElGamal함수가 비대칭 암호(asymmetric cipher)함수이지만 제안한 워터마킹 알고리즘에서는 대칭(symmetric) 함수처럼 사용한다. 일반적인 LSB기법이 가지는 SB 공격을 방어하기 위해 MSB와 LSB의 양쪽 위치를 제외한 나머지 위치에 워터마크를 삽입한다.

제안하는 워터마킹 기법은 워터마크 삽입단계와 검출단계로 구성된다. 삽입 단계에서는 먼저 일방향 해쉬 함수를 사용하여 커버 이미지에서 커버 비트의 위치를 결정한다. 워터마크 데이터의 1비트를 해당 커버 비트 위치에 삽입시키고 워터마크 데이터의 길이만큼 반복 실행한다. 검출 단계에서도 삽입 단계와 마찬가지로 먼저 일방향 해쉬 함수를 사용하여 커버 비트의 위치를 계산해낸다. 해당 위치에 삽입된 워터마크 비트를 읽어들인다.

제안하는 워터마킹 기법의 기본 알고리즘과 가정을 서술하면 다음과 같다.

- 커버이미지 데이터의 크기를 $c_x * c_y$ 라고 하고 c_z를 픽셀당 색상정보라고 하면 커버이미지를 $c_x * c_y * c_z$ 3차원 행렬로 나타낼 수 있다.
- 커버 이미지 데이터의 소유자는 개인키 생성 알고리즘을 사용해 1024비트의 개인키를 K를 가지고 있다고 가정한다.
- 커버 이미지 데이터마다 고유식별자 ID_i를 부여한다.
- 삽입과정과 검출과정은 대칭적 성질이 있다.

[개인키 생성 과정]

1) 1개의 솟수 p를 임으로 선택하고, p보다 작은 2개의 임의의 수 g, x를 선택한다. ($|p| = |g| = 1024$비트라고 가정)
2) $y \equiv g^x \bmod p$값을 구한다.
3) (y, g, p) 값은 공개키가 되고, x는 개인키가 된다.

[삽입과정]

1) 초기 시드(seed)값 s_x, s_y, s_z을 계산한다 : ($|s_x| = |s_y| = |s_z| = 1024$비트)

$$s_x = ID_i^K \bmod p$$
$$s_y = ID_i^{K^2} \bmod p$$
$$s_z = ID_i^{K^4} \bmod p$$

2) 시드값을 가지고 ElGamal 해쉬 함수에 적용시켜 의사 난수 순열 원소를 생성한다 :

$$e_x = s_x y^r \bmod p$$
$$e_y = s_y y^r \bmod p$$
$$e_z = s_z y^r \bmod p$$

3) 의사 난수 순열 원소를 사용하여 커버 데이터 상의 은닉 비트 위치를 계산한다 :

$$l_x = e_x \bmod c_x$$
$$l_y = e_y \bmod c_y$$
$$l_z = e_z \bmod c_z$$

4) 해당 은닉 비트 위치 (l_x,l_y,l_z)에 워터마크 데이터의 1 비트를 삽입한다 :

$$C(l_x,l_y,l_z) \leftarrow W(1)$$

5) 다음 permutation 원소를 계산하기 위한 시드값으로 현재 은닉 비트 위치 인덱스값을 사용한다 :

$$s_x \leftarrow l_x y^r \bmod p$$
$$s_y \leftarrow l_y y^r \bmod p$$
$$s_z \leftarrow l_z y^r \bmod p$$

6) 워터마크 데이터를 모두 삽입할 때까지 2)~5)의 과정을 반복하여 실행한다.

[검출 과정]

1) 초기 시드값 s_x, s_y, s_z을 구한다(삽입 과정의 1단계).
2) 은닉 비트 위치 계산을 위해, 시드값을 ElGamal 해쉬 함수에 적용시켜 의사 난수 순열 원소 p_x, p_y, p_z를 생성한다(삽입 과정의 2단계).
3) 의사 난수 순열 원소를 사용하여 커버 데이터 상의 은닉 비트 위치 l_x, l_y, l_z를 계산한다(삽입 과정 3단계).
4) 해당 은닉 비트 위치 (l_x, l_y, l_z)에 워터마크 데이터의 1비트를 검출한다.
5) 다음 순열 원소를 구하기 위한 시드값을 새로 계산한다(삽입 과정의 5단계).
6) 워터마크 데이터를 모두 검출할 때까지 2)~5)의 과정을 반복 실행한다.

일방향 해쉬 함수를 이용하는 기본적인 삽입 과정의 시드값을 계산하는 단계에서 동일한 해쉬값이 나오는 충돌(collision)이 발생할 수 있다. 이로 인해 의사 난수 순열 원소들 중에 중복이 발생할 수 있으므로 커버 이미지 데이터의 은닉 비트 위치가 중복되어 계산될 수 있다. 결국 은닉 비트 위치의 중복으로 인해 워터마크 데이터의 2개 이상의 비트값들이 커버 이미지 데이터의 1개 비트 위치에 겹쳐서 삽입되게 되므로 워터마크 데이터의 손상을 가져올 수 있다. 해쉬 충돌을 방지하기 위해 해쉬 테이블을 사용한다. 해쉬 인덱스값이 이미지 크기인 n개를 갖도록 구성하며 해쉬테이블에서 중복되는 해쉬값이 발견되는 경우 그대로 삽입시키지 않고 다음 순열 원소를 계산한다.

SB공격으로부터 견고해지기 위해 LSB와 MSB 양쪽 위치를 사용하지 않는다. 커버 이미지 데이터의 은닉 비트 위치를 계산할 때 색상 정보 위치인 lz값이 0과 7인지 검사하여, 해당되지 않는 경우에만 워터마크 비트를 삽입시킨다. 만약 해당되면 역시 삽입시키지 않고 다음 은닉 비트 위치를 계산하기 위해 다음 순열 원소를 계산한다.

4. 구현 및 성능 분석

4.1 구현 및 실험 환경

제안한 워터마킹 알고리즘을 구현하기 위해 Java JDK version 1.2.1을 사용하고 Pentirum-Pro 200 Mhz CPU와 128 MB 메인 메모리가 장착된 PC 환경에서 구현하였다. 암호화에 사용되는 라이브러리는 IAIK 2.51을 사용한다. 커버 이미지 데이터와 워터마크 이미지 데이터로 흑백 256 gray-level의 PGM 파일을 사용한다. 커버 이미지의 크기는 256 * 256이고 워터마크 이미지는 180 * 60 크기를 사용한다.

커버에 워터마크를 삽입할 때 발생기는 노이즈(noise)가 원본 커버 이미지의 화질에 미치는 영향을 평가하기 위해 커버 이미지와 스테고 이미지 사이의 손상율(difference distortion metrics)을 계산한다. 측정 단위로 픽셀의 색상 정보 차이의 평균의 제곱인 MSE와 PSNR을 사용한다. 공격에 대한 워터마킹 기법의 견고성을 평가할 때에도 MSE와 PSNR값을 사용한다[7]. 크기가 동일한 원본 커버 이미지와 워터마크가 삽입된 스테고 이미지 사이의 MSE(mean square error)값과 PSNR(peak signal-to-noise)를 계산하여 2개 이미지 사이의 화질의 손상율을 비교한다.

크기가 n * n인 커버 이미지를 C, 커버 이미지에 워터마크가 삽입된 스테고 이미지를 S 라고 가정한다. C와 S의 (i,j)번째 픽셀의 색상값을 각각 C_{ij}, S_{ij} 라고 표현하면, 2개 이미지사이의 오류 제곱의 평균을 나타내는 MSE은 다음과 같다.

$$MSE = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (S_{ij} - C_{ij})^2$$

C와 S 사이의 화질 손상 정도를 나타내는 PSNR은 다음과 같이 나타낸다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \; [dB]$$

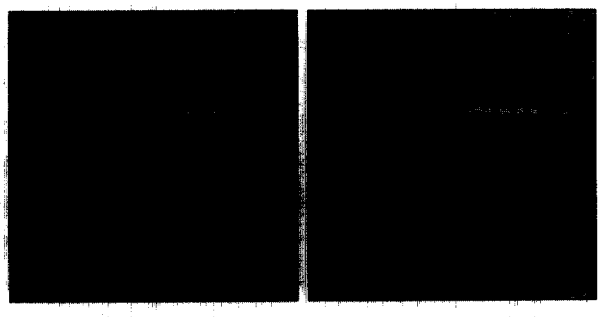
육안으로 원본 이미지에 대한 손상 정도를 식별할 수 있는 PSNR값의 범위를 30dB 이상 60dB 이하라고 가정한다.

4.2 실험 및 성능 분석

제안한 워터마킹 기법의 성능과 공격에 대한 견고성을 분석하기 위해 3가지 실험을 한다. 첫 번째 커버 원본 이미지와 스테고 이미지 사이의 PSNR값을 측정하여 제안한 워터마킹 기법이 가지는 육안 식별 불능성(perceptual invisibility)을 증명하고, 두 번째 스테고 이미지에 대해 흑백 이미지값을 변경시킨 후 공격당한 스테고 이미지로부터 워터마크를 검출하여 손상 상태를 확인하여 감마 보정 공격에 대한 견고성을 검증한다. 세 번째 스테고 이미지에 대한 LSB공격을 수행하여 공격당한 스테고 이미지로부터 추출해낸 워터마크 이미지의 손상 상태를 검사하여 제안한 워터마킹 기법이 가지는 LSB공격에 대한 견고성을 검증한다.

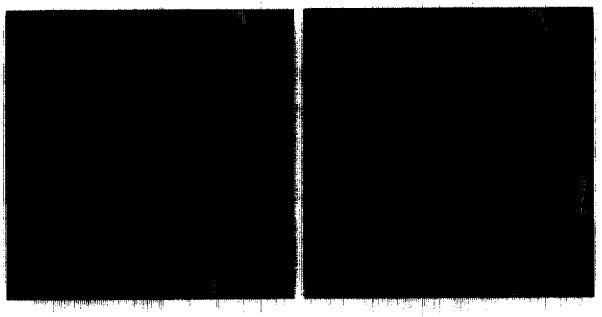
첫 번째 실험을 위해 3가지 종류의 커버 이미지 데이터를 대상으로 워터마크를 삽입시켜 생성한 스테고 이미지의 PSNR값을 측정하였다. 3가지 커버 이미지로 (그림 2)의 (a), (d), (f)를 사용하며 모두 크기가 256 * 256이고 색상은 256 gray scale level을 가진다. 각 커버 이미지에 대해 워터마크를 삽입한 스테고 이미지는 (그림 2)의 (b), (e), (g)가 해당된다. 워터마크 데이터로 사용하는 (그림 2)의 (c)는 영문자 KOREA를 넣은 이미지이고 크기가 180*60이고 256 gray scale level을 가진다. (그림 2.b)는 워터마크를 커버 이미지 (그림 2.a)에 삽입한 스테고 이미지로 PSNR값은 55.46이다. (그림 2.e)는 워터마크를 커버 이미지 (그림 2.d)에 삽입한 스테고 이미지이고 PSNR값은 55.32이다. (그림 2.g)은 워터마크 이미지 (그림 2.c)를 커버 이미지 (그림 2.f)에 삽입한 스테고 이미지이고 PSNR값은 56.11이다.

두 번째 실험으로 비의도적인 워터마크 공격에 대한 견고성을 평가하기 위해 워터마크가 삽입된 스테고 이미지의 흑백 컬러(grayscale)값을 변경시킨 다음, 공격당한 스테고 이미지로부터 워터마크를 추출한다. PGM 파일 형식인 스테고 이미지 (그림 2.e)의 gamma 보정값을 0에서 64로 변경시킨 결과가 (그림 3.a)이다. (그림 3.a)로부터 검출해낸 워터마크 이미지 (그림 3.b)는 오른쪽 기울림의 손상을 입었지만 철자 인식이 가능함을 알 수 있다.

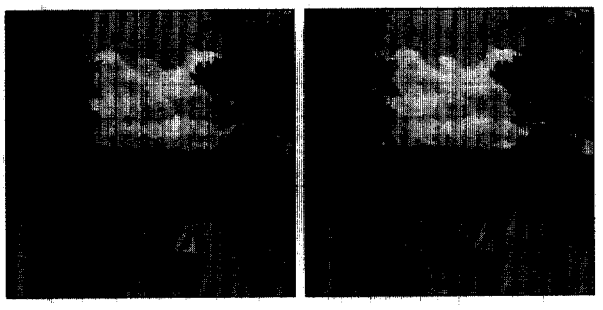


(그림 2) (a) 커버 이미지(KIEL) (b) 그림 2.a의 스테고 이미지 (PSNR = 55.38)

(c) 워터마크 원본

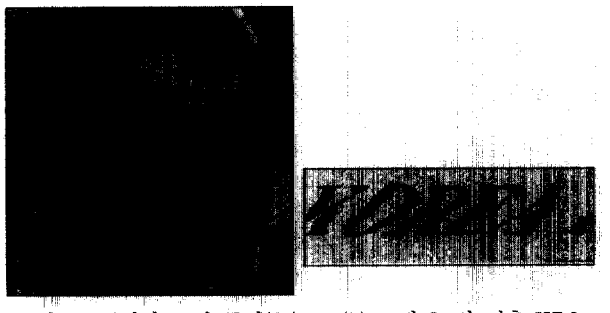


(d) 커버 이미지2(Lenna) (e) 그림 2.d의 스테고 이미지 (PSNR = 56.18)



(f)커버 이미지3(Lake) (g) 그림 2.f의 스테고 이미지 (PSNR = 55.87)

(그림 2)



(그림 3) (a)감마 보정 공격(64) (b) 그림 3.a의 검출 WM

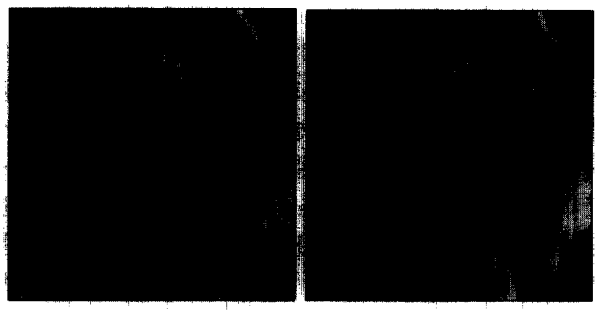
(그림 3)

세 번째 실험으로 워터마크 공격에 대한 견고성을 평가하기 위해 워터마크가 삽입된 스테고 이미지에 LSB공격을 수행한 이미지와 커버 원본 이미지 사이의 PSNR값을 측정한다. 워터마크가 삽입된 스테고 이미지에 대한 LSB공격은 스테고 이미지 바이트 단위로 LSB 위치의 값을 전부 1로 만들어버림으로써 공격당한 이미지의 손상 정도가 가능한 육안으로 판별되지 않는 범위 내에서 이미 삽입되어 있는 워터마크 정보를 정상적으로 추출해내지 못하도록 파괴시키는 데 목적이 있다. 손상된 스테고 이미지의 견고성의 세

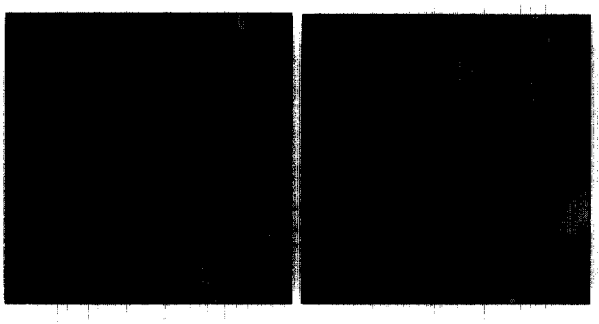
기를 평가하기 위해 LSB의 비트수를 1개부터 4개까지 차례대로 늘려나가면서 공격의 강도를 높인다. 즉, 스테고 이미지의 1바이트(=8비트)당 LSB 공격에서 사용할 비트의 개수, 즉, 원래 값을 지우고 대신 1값으로 채워 넣는 마스크 비트의 개수를 1~4개까지 확대하여 적용시킨다. 공격당한 스테고 이미지로부터 추출해낸 워터마크의 화질 상태들을 비교한다.

LSB공격에 대한 견고성 실험을 위해 원본 커버 이미지로 (그림 2.d)를 사용하고 워터마크로 (그림 2.c)을 사용한다. 워터마크가 삽입된 스테고 이미지 (그림 2.e)에 대해 LSB = 1~4까지 차례대로 LSB공격을 적용시킨 결과가 (그림 4)의 (a), (c), (e), (g)에 나타나 있다. LSB공격당한 스테고 이미지로부터 추출한 워터마크 이미지는 (b), (d), (f), (h)에 나타나 있다.

(그림 4)의 (a)~(g)를 보면 LSB공격의 강도가 더해질수록 공격당한 스테고 이미지의 PSNR값이 떨어지고 공격에 의한 이미지 블러링(blurring)등의 원본 손상이 육안으로 식별가능하여 공격 여부를 감지할 수 있다. (그림 4)의 (b)~(h)을 보면 공격당한 스테고 이미지로부터 추출한 워터마크 이미지는 육안으로 워터마크 글자의 식별이 가능할 정도로 손상되지 않고 견고하다. LSB공격의 목적이 커버 이미지에 삽입된 워터마크 데이터를 손상시켜 추출한 워터마크가 육안으로 식별할 수 없도록 파괴시키는 것이기 때문에, LSB 공격에서 추출한 워터마크 이미지가 육안으로 식별이 가능하다는 것은 LSB공격이 실패했다는 것을 의미한다. 따라서 본 실험을 통해 제안한 워터마킹 기법이 LSB공격에 견고한 안전성을 확인할 수 있다.

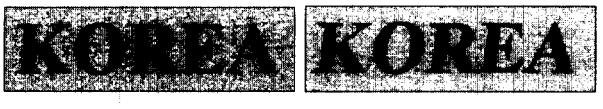


(그림 4) (a)공격1 (PSNR = 48.39) (c) 공격2 (PSNR = 42.83)



(e) 공격3 (PSNR = 37.11) (g) 공격4 (PSNR = 30.25)

(b) 그림 4.a의 검출 WM1 (d) 그림 4.c의 검출WM2



(f) 그림 4.e의 검출WM3 (h) 그림 4.g의 검출WM4

(그림 4)

본 논문에서 제시한 워터마킹 기법과 기존의 LSB방식의 워터마킹 기법들의 성능을 비교하기 위해 각 기법들의 특징을 <표 1>에 제시하였다. 안전한 워터마킹 기법의 조건에 해당하는 6가지 항목을 가지고 비교한다. 워터마킹 기법이 LSB 공격에 대해 가지는 워터마크의 견고성의 세기를 나타내고, 워터마크 검출에 원본 데이터가 필요한지 여부를 비교한다. 워터마킹 기법의 안전성의 근거가 되는 요소를 안전성 인자 항목에 나열한다. 허가 받지 않은 사용자가 적합한 안전성 인자를 사용하지 않고 임의로 워터마크의 위치를 탐지하거나 추출해 내려고 할 때의 성공 난이도를 비교하고, 워터마크가 삽입된 스테고 이미지와 커버 이미지 사이의 육안적 식별이 어렵도록 하는 워터마킹 기법의 육안 식별성(perceputality) 지원 사항을 비교한다. 마지막으로 워터마크를 삽입할 때 사용하는 키 값과 워터마크를 검출할 때 사용하는 키 값이 다른 공개키 워터마킹 방식으로 전환할 수 있는 확장가능성을 비교한다. 비대칭키 방식의 워터마킹 기법은 워터마크 삽입과 검출시 동일한 키가 사용되어야 하기 때문에 모든 허가된 사용자에게 키를 안전하게 전달해야 하는 제약점을 해결할 수 있다. 그런 측면에서 공개키 워터마킹 방식으로의 전환 가능성은 워터마킹 기법의 암호학적 요소의 사용 여부를 의미한다.

〈표 1〉 제안한 워터마킹 기법의 성능 비교 (WM : 워터마크)

비교항목 \ 워터마킹 기법	Aura 방식[1]	Wong 방식[11]	Moller 방식[5]	제안 기법
LSB 공격의 견고성	약함	약함	약함	강함
WM검출시 원본 데이터 사용	×	×	○	×
안전성 인자	키	키	원본 이미지	키
WM탐지의 난이성	높음	높음	낮음	높음
육안 식별의 난이성	높음	높음	높음	높음
공개키 워터마킹 확장성	없음	가능	없음	가능

4.3 안전성 분석

워터마킹 기법의 steganalysis에서 사용되는 공격 방식은 3가지 방식으로 구분된다 : 탐지 공격(detection attack), 손상 공격(destroying attack), 변형 공격(modifying attack)[4, 6]. 본 논문에서 제안한 워터마킹 알고리즘의 안전성을 분

석하기 위해 steganalysis의 3가지 공격에 대한 안전성을 분석한다.

탐지 공격의 경우, 공격자는 스테고 이미지와 일방향 해쉬 알고리즘을 알고 있고 원래 이미지 소유자의 개인키를 모른다고 가정한다. 스테고 이미지에 삽입된 워터마크의 위치를 알아내려면 2가지 방법이 있다. 한가지 방법은 워터마크를 삽입하는데 사용된 일방향 해쉬 함수를 cryptanalysis하여 워터마크의 위치를 직접 알아내는 것이다. 만약 $y = f(x)$ 형태로 일방향 함수를 나타낸다면, 일방향 함수 문제를 풀이한다는 것은 $x = f^{-1}(y)$를 만족하는 $f(x)$의 역함수 $f^{-1}(x)$를 찾아내는 것이다. 이것은 2장에서 서술한바와 같이 계산상으로 불가능하다. 본 논문에서 제안한 워터마킹 기법에서는 ElGamal함수와 모듈라 연산을 사용하였다. 공격자가, 제안한 알고리즘을 사용한 스테고 이미지에서 워터마크를 검출하고자 한다면 ElGamal문제의 역함수를 구해야 한다. ElGamal문제의 안전성은 이산대수문제의 난이도와 동일하다. 따라서 공격자는 일방향 함수를 cryptanalysis하여 워터마크를 검출해낼 수 없다.

다른 한 가지 방법은 이미지 소유자의 개인키 K를 직접 알아내는 것이다. 이를 위해 공격자가 brute-force 공격을 한다고 가정한다. 제안한 워터마킹 기법에 사용되는 개인키의 길이는 1024비트이므로, 2^{1024}가지의 조합이 존재한다. 만약 공격자가 슈퍼 컴퓨터를 사용하여 1년 안에 개인키 K 값을 알아 낼 수 있다면, 그 슈퍼 컴퓨터의 성능은 다음과 같이 반드시 적어도 10^{282} MIPS 이상이어야 할 것이다 :

$$\frac{2^{1024}}{10^{6}*60[\mathrm{sec}]*60[\mathrm{min}]*24[hr]*365[day]} \geq 10^{282}[MIPS]$$

이런 계산 성능을 지닌 컴퓨터는 아직 지구상에 존재하지 않으며 단시일 내에 출현하기 어려울 것으로 보인다. 따라서 공격자는 소유자의 개인키를 알아낼 수 없다.

워터마크를 탐지해내는 또 다른 방법은 다수 개의 스테고 이미지들을 사용하는 것이다. 공격자는 커버 이미지와 스테고 이미지를 임의로 선택할 수 있다고 가정한다. 공격자는 다수의 스테고 이미지를 얻을 수 있고 커버 이미지와 스테고 이미지를 비교할 수 있다. 만약 워터마크 삽입과정에 사용되는 키 값이 동일하다면 공격자는 동일한 크기의 커버 이미지들을 사용하여 얻은 스테고 이미지들로부터 워터마킹의 위치를 알아낼 수 있다. 동일한 크기의 스테고 이미지를 모두 AND 연산시키면 동일한 위치에 존재하는 비트만 1값으로 표시되고 다른 나머지 비트들은 0값으로 표시되기 때문에 동일한 위치에 표시되는 워터마크를 알 수 있다. 제안한 워터마킹 기법에서는 은닉 비트 위치의 집합으로 생성하는 의사 난수 순열 계산을 위한 시드값으로 커버 이미지의 소유자의 개인 키 이외에 커버 이미지에게 부여되는 고유식별자를 사용한다. 커버 이미지마다 생성되는 의사 난수 순열 집합이 서로 다르므로 해당 스테고 이미지에 삽입된 워터마크의 위치가 서로 달라지게 된다. 동일한 크기의 스테고 이미지들을 AND연산시켜도 워터마크를 검출해낼 수 없다. 따라서 제안한 워터마킹 기법은 변형공격에도 안전하다.

손상 공격의 경우, 공격자는 스테고 이미지만을 알고 있다고 가정한다. 공격자는 스테고 이미지에 삽입되어 있는 워터마크를 손상시키기 위해 육안으로 스테고 이미지의 훼손 사실이 판단되지 않도록 LSB나 MSB위치의 비트값들을 모두 0값으로 변경시키거나 임의의 값으로 변경시킬 수 있다. 변형된 스테고 이미지로부터 검출해낸 워터마크는 원래 삽입시켰던 원본 워터마크가 아닌 손상된 상태가 되어 온전한 워터마크를 얻을 수 없게 된다. 제안한 워터마킹 기법에서는 워터마크를 커버 이미지의 LSB나 MSB 위치에 워터마크를 삽입하지 않기 때문에, LSB나 MSB 위치의 비트값을 제거하거나 변형시켜도 이미 삽입된 워터마크는 아무런 영향을 받지 않는다. 따라서 변형된 스테고 이미지로부터 온전한 워터마크를 얻을 수 있다. 만약 제안한 워터마킹 기법을 사용한 스테고 이미지의 워터마크를 손상시키고자 한다면 공격자는 스테고 이미지의 LSB나 MSB가 아닌 모든 SB에 걸쳐 값을 변형시켜야 한다. 이것은 스테고 이미지 전체의 화질에 손상을 가져오게 되고 공격자의 손상 공격 사실이 공개되므로 공격이 실패하게 된다. 따라서 제안한 워터마킹 기법은 손상 공격으로부터 안전하다.

5. 결 론

본 논문에서는 ElGamal 함수를 사용하는 일방향 해쉬 함수에 기반한 디지털 이미지 워터마킹 기법을 제안하였다. 본 논문에서 제안한 워터마킹 기법의 육안적 비 구별성과 견고성을 평가하기 위해 3가지 실험을 수행하였다. 첫 번째 실험 결과로서 워터마크가 삽입된 스테고 이미지와 원본 커버 이미지 사이의 PSNR 값의 범위가 55.xx 사이이므로 이미지의 화질의 차이는 육안으로 구별하기 어렵다는 것을 보였다. 두 번째 실험 결과는 스테고 이미지의 색상 값을 변경시켜 삽입된 워터마크를 손상시켜도 손상된 워터마크가 인식 가능하므로 감마보정 공격에 견고함을 보였다. 세 번째 실험 결과로 인해 제안한 워터마킹 기법은 기존의 LSB공격에 안전한 견고성을 확인하였다.

제안한 워터마킹 기법은 서론에서 서술한 안전한 워터마킹 기법의 조건을 만족한다. ElGamal함수를 사용하는 일방향 해쉬 함수가 생성하는 의사 난수 순열이 은닉 비트 위치를 결정하기 때문에 허가받지 않은 사용자가 은닉 비트의 위치를 알아내는 것은 암호학적인 인수분해 문제에 해당하므로 불가능하다. 워터마크 검출시 원본 이미지 대신 키를 사용한다. 워터마크 비트가 커버 이미지의 SB를 제외

한 임의의 위치에 삽입되기 때문에 다른 워터마킹 기법보다 견고하다. 워터마크 삽입시 커버 원본 이미지 소유자의 개인키를 사용하기 때문에 소유자의 지적 재산권을 보호할 수 있다.

일방향 해쉬 함수를 사용하기 때문에 공격에 대한 견고성을 얻지만 ElGamal함수의 사용은 상당한 계산 시간을 요구하기 때문에 워터마킹 기법 전체의 성능 면에서 비효율적인 인자가 될 수 있다. 따라서 계산 시간이 효율적인 일방향 해쉬 함수를 구현하는 것이 워터마킹 기법의 수행 속도를 향상시킬 수 있을 것이다.

본 논문에서 제시한 워터마킹 기법은 ElGamal함수를 사용하지만 대칭성 워터마킹 방식이다. 대칭성 워터마킹 기법은 워터마크 삽입과 검출시 동일한 키를 사용해야 하기 때문에 허가된 사용자에게 키를 안전하게 배포해야 하는 제약점이 존재한다. 공개키를 사용하는 비대칭성 워터마킹 기법은 대칭성 워터마킹 방식의 단점을 해결할 수 있다. 향후 비대칭성 워터마킹 방식에 대한 연구가 이루어질 수 있을 것이다.

참 고 문 헌

[1] T. Aura, "Practical Invisibility in Digital Communication," Proceedings of Information Hiding Workshop, LNCS Vol. 1174, pp.265-278, Springer-Verlag, 1996.

[2] Bloom, "Copy Protection for DVD Video," Proceedings of IEEE, Vol.87, No.7, July, 1999.

[3] B. Chor, A. Fiat, M. Naor, "Tracing Traitors," Proceedings of Crypto'94, LNCS, Vol.839, pp.257-270, Springer-Verlag, 1994.

[4] S. Craver, N. Memon, B. Yeo, "Resolving Rightful Ownerships with Invisible Watermarking Techni- ques : Limitations, Attacks and Implications," IEEE Journal of SAC, Vol.16, No.4, pp.573-586, May, 1998.

[5] E. Franz, A. Jerichow, S. Moller, A. Pfitzmann, I. Stierand, "Computer Based Steganography : How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense, At Best," Proceedings of Information Hiding Workshop, LNCS Vol.1174, pp.7-21, Springer-Verlag, 1996.

[6] N. F. Johnson, S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software," Proceedings of Information Hiding Workshop, LNCS Vol.1525, pp.273-289, Springer-Verlag, 1998.

[7] S. Katzenbeisser, F. A. P. Petitcolas, editor. Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 1999.

[8] Eugene T. Lin, E. J. Delp, "A Review of Data Hiding in Digital Images," *Proceedings of the Image Processing, Image Quality,Image Capture Systems Conference'99,* pp.274-278, April, 1999.

[9] M. Luby, C. Rackoff, "How to Construct Pseudorandom Permutations from Pseudorandom Functions," SIAM Journal on Computation, Vol.17, No.2, pp.373-386, 1988.

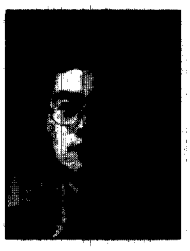
[10] R. C. Merkle, "One-way Hash Functions and DES," Proceedings of Crypto'89, LNCS Vol.435, pp.428-446, Springer-Verlag, 1989.

[11] N. Memon, P. W. Wong, "Protecting Digital Media Content," CACM, Vol.41, No.7, pp.35-43, July, 1998.

[12] M. Naor, O. Reingold, "On the Construction Pseudorandom Permutations : Luby-Rackoff Revisited," Journal of Cryptology, Vol.12, No.1, pp.29-66, 1999.

[13] R. J. Anderson, editor. *Information hiding : first international workshop,* LNCS Vol.1174, Springer-Verlag, May, 1996.

[14] T. ElGamal "A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," Proceedings of Crypto'84, LNCS Vol.196, pp.10-18, Springer-Verlag, 1985.



이 진 호

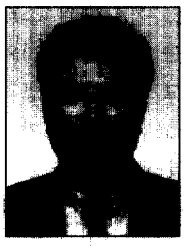
e-mail : jhlee@netlab.korea.ac.kr

1993년 연세대학교 전산학과 졸업(학사)

2001년 고려대학교 컴퓨터학과 졸업(석사)

1996년~1999년 LG-EDS 기술연구소 연구원

관심분야 : 정보보안, 전자상거래, 워터마킹, 네트워크



김 태 윤

e-mail : tykim@netlab.korea.ac.kr

1981년 고려대학교 산업공학과 학사

1983년 미국 Wayne State University 전산과학과 석사

1987년 미국 Auburn University 전산과학과 박사

1988년~현재 고려대학교 컴퓨터학과 교수

관심분야 : 전자상거래, 컴퓨터 네트워크, EDI, 이동통신, 멀티미디어 등