RSA 함수에 기반한 안전한 워터마킹 기법
(A Secure Digital Watermarking Scheme based on RSA Function)

이 진 호 † 김 태 윤 ††
(Jean-Ho Lee) (Tai-Yun Kim)

요 약 디지털 워터마킹(digital watermarking)은 기밀 정보를 디지털 이미지 속에 삽입시켜 이미지 소유자의 저작권을 보호하는 것을 목적으로 하는 기법이다. 저작권 보호를 위한 디지털 워터마킹 기법의 안전성을 보장하기 위해서, 삽입 정보의 위치를 결정할 때 워터마킹 공격에 대한 견고성과 육안적 비구별성을 동시에 추구해야 하고, 워터마킹 알고리즘의 은닉성 대신 키의 은닉성이 보장되어야 하며, 키의 사용으로 허가받지 않은 사용자의 워터마크 검출을 방지할 수 있어야 한다. 이를 위해 본 논문에서는 암호학에서 사용되는 일방향 해쉬 함수를 사용하는 워터마킹 기법을 제안한다. 일방향 해쉬 함수를 구현하기 위해 RSA 일방향 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.

Abstract Digital watermarking is a technique for the purpose of protecting the ownership of the image by embedding invisible watermarks in a digital image. To guarantee the security of the digital watermarking scheme for copyright protection, it is required to satisfy some requirements: robustness and perceptual invisibility which provided by the location of embedded bits, the public watermarking algorithm, and the hidden use of the key, which can protect unauthorized accesses from illegal users. For this, in this paper we propose a new copyright watermarking scheme, which is based on one-way hash functions using RSA functions and modular operations. RSA functions are widely used in cryptographic systems. Our watermarking scheme is robust against LSB(least significant bit) attacks and gamma corresction attack, and is also perceptually invisible. We demonstrate the characteristics of our proposed watermarking scheme through experiments.

1. 서 론

디지털 데이타(digital data) 형태의 특성 중에는 처리 과정의 단순함과 사용의 편리함 그리고 디지털 통신을 통한 대량 배포의 용이함 등의 장점과 데이타가 갖고 있는 내용(content)의 무결성을 보장하기가 어렵다는 단점이 존재한다. 디지털 데이타 형식이 다양한 매체에 널리 적용됨에 따라, 허가받지 않은 접근에 의한 대량 복제나 저작자와 상관없는 무단 변형 등으로 인한 문제점들이 심각하게 나타나고 있다. 디지털 데이타로 저장한 데이타 내용의 저작권을 보호하고 무결성(integrity)을 보장하기 위해 디지털 워터마킹(digital water marking) 기술이 도입되었다[7].

디지털 워터마킹은 디지털 이미지 데이타의 저작권을 보호하고 불법 접근을 방지하는 것을 목적으로 고안되었다. 디지털 워터마킹 기법은 커버(cover)라고 하는 디지털 데이타 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 데이타

• 이 논문은 2000년도 한국학술진흥재단의 지원에 의하여 연구되었음.
† 비 회 원 : 고려대학교 컴퓨터학과
jhlee@netlab.korea.ac.kr
†† 종신회원 : 고려대학교 컴퓨터학과 교수
tykim@netlab.korea.ac.kr
논문접수 : 2000년 12월 2일
심사완료 : 2001년 4월 25일

를 스테고 데이타(stego data)라고 한다. 커버 데이타는 이미지, 오디오, 동영상, 텍스트 등 모든 멀티미디어 데이타를 포괄하며 비밀 정보는 주로 저작권을 나타낼 수 있는 로고나 문구가 사용된다.

저작권 보호를 위한 안전한 디지털 워터마킹 기법이 만족시켜야 하는 필수 조건을 살펴보면 다음과 같다:

1. 원본 커버 이미지와 워터마크가 삽입된 스테고 이미지의 화질이 육안으로 구별할 수 없어야 한다.
② 허가받지 않은 사용자는 워터마크를 검출할 수 없어야 한다.
③ 워터마크를 검출하는데 원본 커버 이미지나 워터마크 이미지를 사용하지 않아야 한다.
④ 워터마크 데이타는 스테고 이미지를 적절한 수준에서 손상시키는 이미지 처리에 견고해야 한다.
⑤ 워터마킹 기법 알고리즘은 공개되어야 하고 오직 개인키의 은닉성이 이미지 저작권을 보호할 수 있다.

워터마킹 기법중 커버 이미지의 LSB(least significant bit)들만을 사용하여 정보를 삽입하는 LSB 방식에 일방향 해쉬 함수(one-way hash function)를 이용하는 방식이 Aura에 의해 도입되었다[1]. Aura는 커버 비트의 인덱스 원소로 구성된 의사 난수 순열(pseudo-random permutation) 집합을 형성함으로써 커버 이미지의 은닉 비트 위치를 결정하였다. 커버 이미지 상의 비트 인덱스 대해 의사 난수 생성기(pseudo-random generator)를 적용하여 의사 난수 순열 집합을 생성하였다. 의사 난수 순열 집합의 원소에 대해 모듈라 연산을 수행하여 은닉 비트 후보 인덱스 집합을 생성하였다. 의사 난수 생성기로 Luby-Rackoff 방식[9,12]을 채택하고 블록 암호(block cipher)함수로 SHS-1을 사용하고 모듈라 연산을 함께 사용하여 일방향 해쉬 함수를 구현하였다. 그러나 Aura 기법도 기존의 LSB방식의 디지털 워터마킹 기법처럼, 중요 비트(significant bit) 공격에 약하다. 즉, 공격자(attacker)가 LSB이나 MSB 값을 모두 0으로 만들어 버리면 워터마크를 복구할 수 없게 된다.

본 논문에서는 Aura 기법과 같이 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 사용하였다. 일방향 해쉬 함수를 구현하기 위해 일방향 함수와 모듈라 연산을 함께 사용한다. 본 논문에서 제안하는 워터마킹 기법은 LSB와 MSB를 제외한 SB(significant bit)를 이용한다. 따라서 기존의 LSB방식의 워터마킹 기법과 달리 LSB 공격에 견고하다. 제안한 워터마킹 기법은 기존의 LSB 공격에 대해 견고하며 안전한 워터마킹 기법의 조건을 만족시킨다.

본 논문의 구성은 다음과 같다. 2장에서 기존 디지털 이미지 워터마킹 기법의 특징을 고찰하고 암호학적 요소인 RSA문제와 일방향 해쉬 함수에 대해 살펴본다. 3장에서 새로운 디지털 이미지 워터마킹 기법을 제안하고, 4장에서 안전성을 분석한다. 5장에서 실험을 통해 제안한 알고리즘의 성능을 평가하고 6장에서 결론 및 향후 과제를 제시한다.

2. 관련 연구

2.1 기존의 워터마킹 기법

디지털 이미지 워터마킹 기법을 커버 이미지에 워터마크를 삽입하는 방식에 따라, LSB 삽입 기법, 도메인 전환(domain transform) 기법, 인지 마스킹(perceptual masking) 기법 3가지로 분류할 수 있다. LSB기법은 공간 영역에서 사용되는 가장 단순한 방식으로, 커버 이미지를 구성하는 픽셀(pixel) 정보를 대상으로 픽셀 단위의 데이타의 LSB 위치에 워터마크 데이타 1개 비트를 직접 삽입시키는 기법이다[5]. 육안으로 감별하기 어렵게 만들기 위해 의사 난수 순열을 사용하여 커버 이미지의 은닉 비트를 선택하기도 한다[1]. LSB기법은 구현이 쉽고 육안 구별이 어려운 장점이 있지만, 모든 LSB를 0값으로 만들어 버리는 LSB공격에 대해 치명적인 약점이 있다. 도메인 전환 기법은 DCT(Discrete Cosine Transform), DFT(Discrete Fourier Transform), DWT(Discrete Wavelet Transform)와 같은 전환 함수를 사용하여 커버 데이타와 워터마크 데이타를 공간 영역(spatial domain)에서 주파수 영역(frequency domain)으로 전환시켜 얻은 계수(coefficient)값을 연산함으로써 워터마크를 삽입한다. 도메인 전환 기법은 압축이나 절단(cropping)같은 영상처리 공격에 대해 견고성이 높다. 인지 마스킹 기법은 커버 이미지 영역에서 육안으로 구별가능한 영역을 결정하여 그 부분에 워터마크를 삽입하는 기법으로 공간 도메인과 전환 도메인 양쪽에서 모두 적용가능하다[8].

저작권 보호를 위해 워터마크를 삽입시키는 경우, 저작자만이 워터마크를 추출해낼 수 있으므로 무단 복제에 의한 원본 저작자에 대한 권리침해를 방지할 수 있다[11]. 전자상거래 환경에서 디지털 상품을 판매하는 경우 디지털 상품마다 유일한 워터마크를 삽입시켜서 구매자 식별이 가능하므로 불법복제에 의한 불법유통의 근원지를 추적할 수 있다[3]. DVD 시스템에서처럼 특

수 하드웨어 장치와 함께 사용되는 경우 워터마크에 복사 계수를 포함시켜서 복사 회수를 관리하여 무단 복제를 방지할 수 있다[2]

2.2 일방향 해쉬 함수

본 논문에서 제안하는 워터마킹 기법은 Aura의 개념처럼 일방향 해쉬 함수에 기반한다. 암호화(encryption) 함수가 일방향 해쉬 함수를 구현할 수 있다는 Merkle의 주장[10]에 따라, 일방향 해쉬 함수를 구현하기 위해 대표적인 암호화 함수인 RSA함수를 사용한다.

제안하는 워터마킹 기법의 일방향 해쉬 함수는 RSA 함수[14]와 모듈라 연산을 이용하여 구현한다.

일방향 해쉬 함수는 4가지 성질을 만족하는 함수 f:x →y 로 정의할 수 있다[10]:

① 함수 f는 모든 크기의 인자에 적용될 수 있다.

② 함수 f는 고정된 크기의 출력을 만들어 낸다.

③ 함수 f, 값 x가 주어질 때, f(x)값은 쉽게 계산된다.

④ 함수 f가 주어질 때, f(x)=f(x′)이면서 x≠x′을 만족하는 x와 x′을 찾는 것은 계산상 불가능하다.

RSA함수는 암호학적인 기본 요소로서 암호화(encryption) 기법과 복호화(decryption) 기법이 존재한다. 먼저 2개의 솟수 p,q를 선택하여 n=p*q 값을 계산한다. gcd(d, (p-1)(q-1))=1을 만족하는 정수 d를 선택하여 e*d ≡ 1 mod (p-1)(q-1)을 만족하는 d의 역원인 정수 e값을 확장 유클리디안 알고리즘을 이용해 계산한다.

평문 메시지를 M, M을 암호화한 암호문을 C라고 표시한다.

◇ 암호화 기법 : 메시지 M을 암호문 C로 암호화시키기 위해 $C \equiv M^e \pmod n$을 계산한다.

◇ 복호화 기법 : 암호문 C를 복호화시키기 위해 $M' \equiv C^d \pmod n$을 계산한다.

$$M' \equiv C^d \pmod n \equiv (M^e)^d \pmod n \equiv M^{k(p-1)(q-1)+1} \pmod n \equiv M \pmod n$$

RSA함수를 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 데이타의 은닉 비트 집합을 생성하기 위해 RSA함수와 모듈라 연산을 사용하여 일방향 해쉬 함수를 구현한다. 의사 난수 순열을 얻기 위해 RSA함수를 사용하고, RSA함수가 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 데이타의 1비트를 해당 커버 비트 위치에 삽입시키고 워터마크 데이타의 길이만큼 반복 실행한다. 검출 단계에서도 삽입 단계와 마찬가지로 먼저 일방향 해쉬 함수를 사용하여 커버 비트의 위치를 계산해 낸다. 해당 위치에 삽입된 워터마크 비트를 읽어들인다.

제안하는 워터마킹 기법의 기본 알고리즘과 가정을 서술하면 다음과 같다.

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

[개인키 생성 과정]

1) 2개의 솟수 p,q를 임으로 선택하고 n=p*q값을 계산한다. (|p|=|q| = 256비트라고 가정)
2) e*d = 1 mod (p-1)(q-1)을 만족하는 e,d 값을 구한다.
3) (n,e) 값을 공개한다.

[삽입과정]

1) 초기 시드(seed)값 s_x,s_y,s_z을 계산한다: ($|s_x|=|s_y|=|s_z|$=512비트)

$$s_x = ID_i^{K} \bmod n$$

$$s_y = ID_i^{K^2} \bmod n$$

$$s_z = ID_i^{K^4} \bmod n$$

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

$$p_x = (s_x)^e \bmod n$$

$$p_y = (s_y)^e \bmod n$$

$$p_z = (s_z)^e \bmod n$$

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

$$l_x = p_x \bmod c_x$$

$$l_y = p_y \bmod c_y$$

$l_z = p_z \bmod c_z$

4) 해당 은닉 비트 위치 (l_x,l_y,l_z)에 워터마크 데이타의 1비트를 삽입한다:
$C(l_x,l_y,l_z) \leftarrow W(1)$
5) 다음 permutation 원소를 계산하기 위한 시드값으로 현재 은닉 비트 위치 인덱스값을 사용한다:
$s_x \leftarrow (l_x)^e \bmod n$
$s_y \leftarrow (l_y)^e \bmod n$
$s_z \leftarrow (l_z)^e \bmod n$
6) 워터마크 데이타를 모두 삽입할 때까지 2)~5)의 과정을 반복하여 실행한다.

[검출 과정]

1) 시드값 s_x, s_y, s_z 을 구한다.
2) 시드값을 RSA 해쉬 함수에 적용시켜 의사 난수 순열 원소 p_x, p_y, p_z 를 생성한다.
3) 의사 난수 순열 원소를 사용하여 커버 데이타 상의 은닉 비트 위치 l_x, l_y, l_z 를 계산한다.
4) 해당 은닉 비트 위치 (l_x, l_y, l_z)에 워터마크 데이타의 1비트를 검출한다.
5) 다음 순열 원소를 구하기 위한 시드값을 새로 계산한다.
6) 워터마크 데이타를 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 데이타의 은닉 비트 위치가 중복되어 계산될 수 있다. 결국 은닉 비트 위치의 중복으로 인해 워터마크 데이타의 2개 이상의 비트값들이 커버 이미지 데이타의 1개 비트 위치에 겹쳐서 삽입되게 되므로 워터마크 데이타의 손상을 가져올 수 있다. 해쉬 충돌을 방지하기 위해 해쉬 테이블을 사용한다. 해쉬 인덱스값이 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 데이타의 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. 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 분석하기 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 RSA함수와 모듈라 연산을 사용하였다. 공격자가, 제안한 알고리즘을 사용한 스테고 이미지에서 워터마크를 검출하고자 한다면 RSA문제의 역함수를 구해야 한다. RSA문제의 안전성은 인수분해 문제의 난이도와 동일하다. 따라서 공격자는 일방향 함수를 cryptanalysis하여 워터마크를 검출해 낼 수 없다.

다른 한가지 방법은 이미지 소유자의 개인키 K를 직접 알아내는 것이다. 이를 위해 공격자가 brute-force 공격을 한다고 가정한다. 제안한 워터마킹 기법에 사용되는 개인키의 길이는 1024비트이므로, 공격자가 1000 MIPS 성능의 컴퓨터를 사용하여 개인키값을 계산한다고 가정하면 소요되는 계산 시간은 다음과 같다:

$$\frac{2^{1024}}{1000 * 10^6 * 60[\text{sec}] * 60[\text{min}] * 24[hr] * 365[day]} \geq 10^{288}[year]$$

이것은 현실적으로 실현 불가능함을 의미한다. 따라서 공격자는 소유자의 개인키를 알아낼 수 없다.

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 없

게 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 가져 오게 되고 공격자의 손상 공격 사실이 공개되므로 공격이 실패하게 된다. 따라서 제안한 워터마킹 기법은 손상 공격으로부터 안전하다.

변형 공격의 경우, 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 알수 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.

5. 구현 및 성능 분석

5.1 구현 및 실험 환경

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

커버에 워터마크를 삽입할 때 발생되는 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:

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

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.

5.2 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 데이타를 대상으로 워터마크를 삽입시켜 생성한 스테고 이미지의 PSNR값을 측정하였다. 3가지 커버 이미지로 <그림1>의 (a), (d), (f)를 사용하며 모두 크기가 256*256 이고 색상은 256 gray scale level을 가진다. 각 커버 이미지에 대해 워터마크를 삽입한 스테고 이미지는 <그

림1>의 (b), (e), (g)가 해당된다. 워터마크 데이타로 사용하는 <그림1>의 (c)는 영문자 KOREA를 넣은 이미지이고 크기가 180*60이고 256 gray scale level을 가진다. <그림 1.b>는 워터마크를 커버 이미지 <그림 1.a>에 삽입한 스테고 이미지로 PSNR값은 55.46이다. <그림 1.e>는 워터마크를 커버 이미지 <그림 1.d>에 삽입한 스테고 이미지이고 PSNR값은 55.32이다. <그림 1.g>은 워터마크 이미지 <그림 1.c>를 커버 이미지 <그림 1.f>에 삽입한 스테고 이미지이고 PSNR값은 56.11이다.

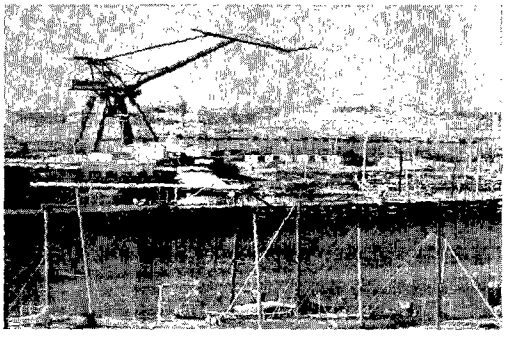
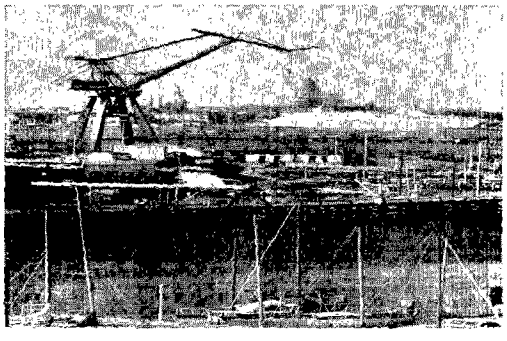


그림 1 (a) 커버 이미지(KIEL)



(b) 그림 1.a의 스테고 이미지 (PSNR= 56.46)

(c) 워터마크 원본

(d) 커버 이미지2 (lenna)

(e) 그림 1.d의 스테고 이미지 (PSNR=55.32)

(f) 커버 이미지3 (lake)

(g) 그림 1.f의 스테고 이미지 (PSNR=56.11)

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 1.e>의 gamma 보정값을 0에서 64로 변경시킨 결과가 <그림 2.a>이다. <그림 2.a>로부터 검출해낸 워터마크 이미지 <그림 2.b>는 오른쪽 기울림의 손상을 입었지만 철자 인식이 가능함을 알 수 있다.

그림 2 (a) 감마 보정 공격(64)

(b) 그림 2.a의 검출 WM

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 세기를 평가하기 위해 LSB의 비트수를 1개부터 4개까지 차례대로 늘려나가면서 공격의 강도를 높인다. 즉, 스테고 이미지의 1바이트(=8비트)당 LSB 공격에서 사용할 비트의 개수, 즉, 원래값을 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 1.a>를 사용하고 워터마크로 <그림 1.c>을 사용한다. 워터마크가 삽입된 스테고 이미지 <그림 1.b>에 대해 LSB=1~4까지 차례대로 LSB공격을 적용시킨 결과가 <그림 3>의 (a), (c), (e), (g)에 나타나 있다. LSB공격당한 스테고 이미지로부터 추출한 워터마크 이미지는 (b), (d), (f), (h)에 나타나 있다.

<그림 3>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. <그림3>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 데이타를 손상시켜 추출한 워터마크가 육안으로 식별할 수 없도록 파괴시키는 것이기 때문에, LSB공격에서 추출한 워터마크 이미지가 육안으로 식별이 가능하다는 것은 LSB공격이 실패했다는 것을 의미한다. 따라서 본 실험을 통해 제안한 워터마킹 기법이 LSB공격에 견고한 안전성을 확인할 수 있다.

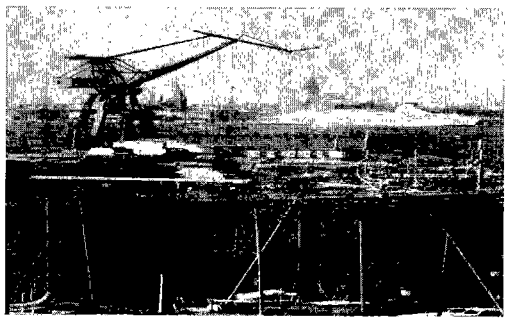
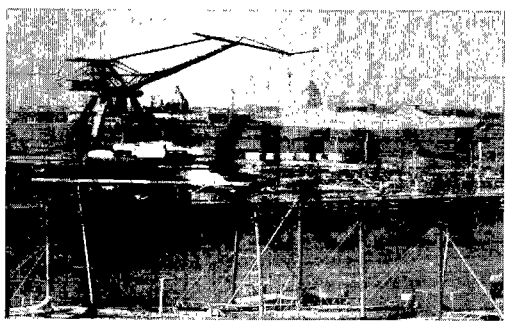
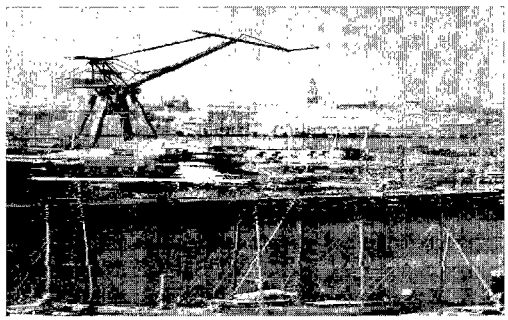


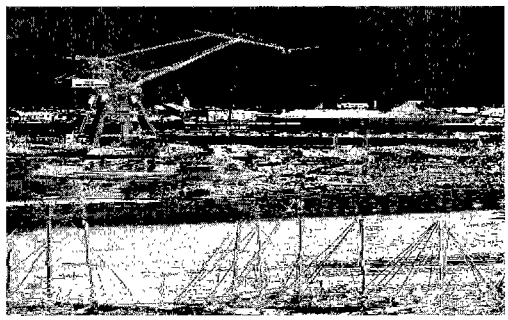
그림 3 (a) 공격1 (PSNR=49.73)



(c) 공격2 (PSNR=43.19)



(e) 공격3 (PSNR=36.24)



(g) 공격4 (PSNR=30.82)

(b)그림 3.a의 검출 WM1

(d) 그림 3.c의 검출 WM2

(f) 그림 3.e의 검출 WM3

(h) 그림 3.g의 검출 WM4

본 논문에서 제시한 워터마킹 기법과 기존의 LSB방식의 워터마킹 기법들의 성능을 비교하기 위해 각 기법들의 특징을 표1에 제시하였다. 안전한 워터마킹 기법의 조건에 해당하는 6가지 항목을 가지고 비교한다. 워터마킹 기법이 LSB 공격에 대해 가지는 워터마크의 견고성의 세기를 나타내고, 워터마크 검출에 원본 데이타가 필요한지 여부를 비교한다. 워터마킹 기법의 안전성의 근거가 되는 요소를 안전성 인자 항목에 나열한다. 허가받지 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 기법과 기존의 워터마킹 기법들과의 비교 (WM:워터마크)

워터마킹 기법 / 비교항목	Aura 방식[1]	Wong 방식[11]	Moller 방식[5]	제안 기법
LSB 공격의 견고성	약함	약함	약함	강함
WM검출시 원본 데이타 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	가	없음	가능

6. 결 론

본 논문에서는 RSA함수를 사용하는 일방향 해쉬 함수에 기반한 디지털 이미지 워터마킹 기법을 제안하였다. 본 논문에서 제안한 워터마킹 기법의 육안적 비구별성과 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 스

테고 이미지의 색상값을 변경시켜 삽입된 워터마크를 손상시켜도 손상된 워터마크가 인식가능하므로 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. RSA함수를 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 제외한 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 RSA함수의 사용은 상당한 계산 시간을 요구하기 때문에 워터마킹 기법 전체의 성능면에서 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 RSA함수를 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 Techniques: 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] R.Rivest, A.Shamir, L.Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," CACM, Vol.21, No.2, pp.120-126, Feb. 1978.

이 진 호
1993년 연세대학교 전산학과 졸업. 1996년 ~ 1999년 LG-EDS 기술연구소 근무. 1999년 ~ 현재 고려대학교 컴퓨터학과 석사과정. 관심분야는 컴퓨터 네트워크, 이동통신, 네트워크 보안, 암호학, 전자상거래

김 태 윤
정보과학회논문지 : 컴퓨팅의 실제 제 7 권 제 1 호 참조