Journal of the Korea Academia-Industrial cooperation Society
Vol. 17, No. 11 pp. 196-201, 2016

http://dx.doi.org/10.5762/KAIS.2016.17.11.196
ISSN 1975-4701 / eISSN 2288-4688

디지털 영상의 효율적인 변형 검출에 관한 연구

우찬일[1*], 이승대[2]
[1]서일대학교 정보통신과, [2]남서울대학교 전자공학과

A Study on Efficient Tamper Detection of Digital Image

Chan-Il Woo[1*], Seung-Dae Lee[2]
[1]**Dept. of Information and Communication Engineering, Seoil University**
[2]**Dept. of Electronic Engineering, Namseoul University**

요 약 디지털 워터마킹은 디지털 미디어 내부에 정보를 숨기기 위해 사용하는 기술로, 디지털 워터마킹 기술은 Robust 워터마킹 기술과 Fragile 워터마킹 기술로 분류할 수 있다. Robust 워터마킹 기술은 일반적으로 저작권을 보호하기 위한 목적으로 사용되고 있으며, Fragile 워터마킹 기술은 영상에 대한 인증과 무결성을 검증하기 위한 목적으로 사용된다. 따라서 Fragile 워터마킹 기술에서는 워터마킹 된 영상에 대해 작은 변형이라도 발생 하였을 경우 삽입된 워터마크는 쉽게 부수어져야 되는 특성을 가지고 있어야 한다. 본 논문에서는 공간 영역에서 영상의 변형 검출을 위한 효과적인 Fragile 워터마킹 방법을 제안하였으며, 본 논문에서 제안한 방법에서는 해쉬 코드와 대칭키 암호를 사용하였다. 제안 방법에서는 원 영상을 작은 크기로 구성된 여러 블록들로 분할한 후 워터마크를 삽입하기 때문에 cut and paste와 같은 공격에 취약하지 않은 특징을 가지고 있으며, 변형 검출을 위하여 워터마크가 삽입된 영상에 대해 전체 블록을 검사하지 않고도 변형이 발생된 부분만을 빠르게 검출할 수 있는 장점이 있다.

Abstract Digital watermarking is a technique used to hide information within digital media. Digital watermarking techniques can be classified as either robust watermarking or fragile watermarking. Robust watermarking techniques are generally used for the purpose of copyright protection. In addition, fragile watermarking techniques are used for the authentication and integrity verification of a digital image. Therefore, fragile watermarks should be easily breakable for trivial tampering of a watermarked image. This paper proposes an efficient fragile watermarking method for image tamper detection in the spatial domain. In the proposed method, a hash code and symmetric key encryption algorithm are used. The proposed method of inserting a watermark by dividing the original image into many blocks of small sizes is not weak against attacks, such as cut and paste. The proposed method can detect the manipulated parts of a watermarked image without testing the entire block of the image.

Keywords : Authentication, Encryption, Fragile Watermarking, Hash Function, Tamper Detection

1. 서론

디지털로 표현된 영상이나 음성과 같은 멀티미디어 데이터는 아날로그 데이터에 비하여 잡음에 강인한 장점이 있으나 디지털 데이터는 복사와 조작이 용이한 단점이 있어 불법적인 복사나 조작으로부터 저작권을 보호하기 위하여 디지털 워터마킹 기술이 개발되었다. 디지털 워터마킹 기술은 디지털 영상 등에 저작권자의 정보인 워터마크를 삽입하여 저작권에 대한 분쟁이 발생하였을 경우 삽입된 워터마크를 추출하여 법적 분쟁의 증거 자료로 사용할 수 있다. 이와 같이 디지털 워터마킹은 저작권을 보호하기 위한 용도로 개발되어 현재까지 다양한

본 논문은 2016년도 서일대학교 학술연구비에 의해 연구되었음.
*Corresponding Author : Chan-Il Woo(Seoil Univ.)
Tel: +82-2-490-7556 email: ciwoo@seoil.ac.kr
Received August 10, 2016 Revised October 4, 2016
Accepted November 10, 2016 Published November 30, 2016

방법들이 제안되고 있으나 저작권을 보호하기 위해서는 필터링이나 압축과 같은 공격에도 삽입된 워터마크를 추출할 수 있어야 한다. 그러나 다양한 공격에 대하여 워터마크를 완벽하게 추출하기에는 기술적으로 어려운 문제점을 가지고 있다. 따라서 저작권 보호를 위한 워터마킹 기술은 워터마크를 제거하기 위한 공격에 강인한 특성을 가질 수 있도록 기술 개발이 진행되고 있다.

워터마킹 기술은 저작권 보호뿐만 아니라 멀티미디어 데이터에 대한 조작 여부를 확인하기 위한 무결성 검증 목적으로도 연구되고 있다. 디지털 영상에 대한 무결성을 검증하기 위한 워터마킹에서는 워터마크가 삽입된 영상에 대하여 작은 변화라도 발생할 경우 삽입된 워터마크가 쉽게 부수어지는 특성을 가지고 있어야 한다[1-3]. 이러한 특성은 저작권을 보호하기 위한 기술과 반대되는 특성으로 저작권 보호를 위한 기술은 워터마크의 강인성을 위해 주파수 영역에서 워터마크를 삽입하지만, 무결성을 검증하기 위한 방법에서는 주파수영역뿐만 아니라 공간영역에서도 워터마크를 삽입하고 있다. 공간영역에서 워터마크를 삽입할 경우 화질 저하를 최소화하기 위해 일반적으로 화소의 하위 두 개 이내의 LSB에 워터마크를 삽입한다[4-6]. 공간영역 방법은 LSB를 제거하는 공격에 취약한 단점이 있으나 워터마크가 삽입된 영상에 대한 변형 여부와 위치를 효과적으로 검출할 수 있는 장점이 있다. 본 연구에서는 변형이 발생된 화소를 검출하기 위한 기존의 방법을 개선하여 변형이 발생된 부분을 4×4 블록 단위로 검출할 수 있는 새로운 방법을 제안한다.

2. 워터마크 삽입을 위한 영상 구조

워터마크를 삽입하기 위한 영상 구조는 기존에 제안한 방법[6]을 보완하여 원 영상을 4개의 영역으로 분할하고, 분할된 영역을 다시 하위 4개의 영역으로 분할한다. 이와 같이 과정을 반복적으로 수행하게 되면 분할된 블록의 최소 크기는 4×4 크기가 되고 이때 블록의 분할을 종료한다. Fig. 1은 원 영상을 4×4 크기의 블록이 생성될 때까지 Level 1의 한개 블록을 분할하는 과정을 나타내고 있다. 그림에 나타낸 바와 같이 분할된 하위 4개의 블록들은 하나의 상위 블록에 포함된다.

본 논문에서는 분할되지 않은 128×128 크기의 원 영상을 Level 0이라고 정의하고 원 영상을 4개의 서브 블록으로 분할한 것을 Level 1이라고 정의한다. Level 1에서 분할된 4개의 블록들 중 하나의 블록을 다시 분할하게 되면 4개의 32×32 블록들이 생성되고 이것을 Level 2라고 정의한다. Level 1의 블록은 총 4개 이므로 4개의 블록들을 모두 분할하게 되면 Level 2에서는 총 16개의 서브블록들이 생성된다. 따라서 각 Level에서 하나의 블록을 분할하게 되면 그 블록의 1/4 크기에 해당하는 4개의 하위 블록이 생성되고, 생성된 하위 블록들은 하나의 상위 블록에 포함된다. 이와 같은 과정으로 원 영상을 분할하게 되면 분할되는 블록들의 크기는 점점 작아지게 되고 최종적으로 4×4 블록이 생성될 때까지 분할된다.

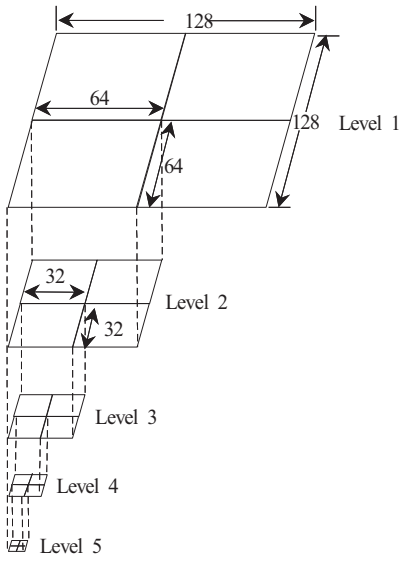

Fig. 1. Divided image

본 논문에서는 워터마크 삽입을 위해 Table 1에 나타낸 것과 같이 원 영상인 Level 0부터 최종 분할된 영상인 Level 5까지 각 Level에 해당하는 블록들을 구해야 한다.

Table 1. The block size of each level

	Block Size	The number of blocks
Level 0	128 × 128	1
Level 1	64 × 64	4
Level 2	32 × 32	16
Level 3	16 × 16	64
Level 4	8 × 8	256
Level 5	4 × 4	1,024

즉, Level 2에서는 16개의 32×32 블록을 구해야하고 Level 3에서는 64개의 16×16 블록을 구해야한다. 이와 같은 구조로 워터마크를 삽입하는 이유는 전체 영상을 검사하지 않고도 상위 블록에 대한 변형이 발생하였을 경우에만 다음의 방법으로 하위 블록들을 검사하여 변형이 발생된 블록들을 4×4 단위로 검출하기 때문이다.

단계 1 : Level 1의 4개의 블록들 중 변형이 발생한 블록을 검사한다. 만약 변형이 발생한 블록이 검출될 경우 변형이 발생된 모든 블록을 검사할 때까지 단계 2부터 5까지의 과정을 반복한다.

단계 2 : Level 1에서 변형이 발생한 블록에 포함되는 4개의 Level 2 블록들에 대하여 변형 여부를 검사한다. 만약 변형이 발생한 블록이 검출될 경우 변형이 발생된 모든 블록을 검사할 때까지 단계 3부터 5까지의 과정을 반복한다.

단계 3 : Level 2 블록들 중 변형이 발생한 블록에 포함되는 Level 3 블록에 대한 변형 여부를 검사한다. 만약 변형된 블록이 검출될 경우 변형이 발생한 모든 블록을 검사할 때까지 단계 4부터 5까지의 과정을 반복한다.

단계 4 : Level 3 블록들 중 변형이 발생한 블록에 포함되는 Level 4 블록에 대한 변형 여부를 검사한다. 만약 변형이 발생된 블록이 검출될 경우 단계 5를 수행한다.

단계 5 : Level 4 블록들 중 변형이 발생된 블록에 포함되는 Level 5 블록에 대한 변형 여부를 검사하여 변형이 발생된 4×4 블록을 검출한다.

기존 방법에서는 변형된 부분을 16×16 블록 단위로 검출할 수 있었지만, 본 논문에서 제안하는 방법에서는 원 영상에 변형이 발생하였을 경우 단계 1부터 단계 5까지의 과정을 수행하면 모든 4×4 블록들을 검사하지 않고도 변형된 부분을 4×4 블록 단위로 검출할 수 있는 장점이 있다.

3. 워터마크 구조

워터마크 삽입을 위한 제안된 영상 구조에서는 Level 3 블록들의 하위 2개의 비트에 워터마크를 삽입한다. Level 3의 각 블록들은 256개의 LSB로 구성되어 있기 때문에 각 블록의 LSB에는 Level 3과 연관되는 Level 0부터 Level 4까지의 워터마크를 삽입한다. 이때 LSB에 삽입되는 각 Level의 워터마크들은 8×8 크기로 변형이 발생한 블록들을 검출한다. 그리고 하위 두 번째 LSB에는 변형이 발생한 8×8 블록들에 대하여 최종적으로 4×4 블록 단위로 변형이 발생된 블록을 검출하기 위한 워터마크를 삽입한다. 즉, Level 3 블록의 LSB에는 8×8 블록 단위로 오류를 검출하기 위한 워터마크를 삽입하고, 두 번째 LSB에는 변형이 발생된 부분을 4×4 블록 단위로 검출하기 위한 워터마크를 삽입한다.

Level 4와 Level 5의 경우, 8×8 블록과 4×4 블록으로 구성되어 있기 때문에 Level 4와 Level 5의 블록들은 각각 64개와 16개의 LSB로 구성되어 있다. 따라서 워터마크가 삽입된 영상에 대한 변형을 검출하기 위한 상위 및 하위 Level의 워터마크를 삽입하기에는 공간이 부족하다. 따라서 워터마크 삽입을 위한 공간 확보를 위하여 Level 3의 블록들에 워터마크를 삽입한다.

3.1 워터마크 생성

제안 방법에서는 전체 영상에 대한 변형 여부를 검사하여 변형이 발생하였다고 판단되면 전체 영상을 Fig. 1의 Level 1과 같이 워터마크가 삽입된 영상을 4개의 블록으로 분할하여 각 블록의 변형 여부를 검사한 후 변형이 발생하지 않았다고 판단되면 해당 영역에 대한 검사는 종료하고 다른 블록을 검사한다. 그러나 변형이 발생된 블록이라고 판단되면 해당 블록에 포함되는 Level 2의 블록들에 대한 변형 여부를 검사하고 변형이 발생된 블록에 대해서만 Level 3의 블록들을 검사한다. 이와 같은 과정으로 Level 5까지 검사하면 최종적으로 변형이 발생된 블록을 4×4 단위로 검출할 수 있다.

따라서 이와 같은 과정으로 변형 여부를 검출하기 위해서는 영상의 변형 여부를 효율적으로 검출할 수 있는 정보를 생성해야 한다. 본 논문에서는 일방향 해쉬 함수(MD5)와 대칭키 암호[7]를 사용하여 변형 여부를 검출하기 위한 워터마크를 생성한다. 해쉬 함수는 입력 데이터에서 하나의 비트라도 서로 다를 경우 그때 생성되는 출력은 매우 다른 값을 생성하는 특성이 있어 무결성 검증에 효율적으로 사용되고 있다. 일반적으로 무결성 검증을 위해서는 해쉬 함수와 공개키 암호를 사용한다. 그러나 RSA와 같은 공개키 암호를 사용할 경우 2,048 비

트 이상의 키를 사용해야 안전성을 보장 받을 수 있다. 그러나 RSA 공개키 암호에서는 매우 큰 두 소수의 곱으로 나눈 나머지 값을 암호문으로 사용하는데 이러한 연산을 수행할 경우 생성되는 암호문의 크기가 매우 커질 수 있어 256개의 비트만 저장할 수 있는 LSB에 삽입하지 못하는 문제가 발생할 수 있다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 대칭키 암호를 사용한다.

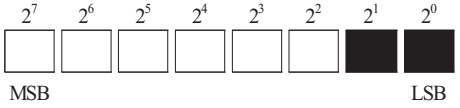

Fig. 2. The watermark insertion location

Fig. 2는 각 화소에서 워터마크가 삽입되는 위치를 나타내고 있다. 그림에서 LSB에 삽입되는 워터마크는 Level 0부터 Level 4까지의 워터마크를 삽입하기 위한 공간으로 사용되어 8×8 블록 단위의 변형 위치를 검출한다. 그리고 하위 두 번째 LSB는 4×4 블록 단위로 변형 위치를 검출하기 위한 정보를 삽입하기 위한 공간으로 사용된다. 즉, LSB는 Level 4까지의 변형 검출을 위한 정보를 삽입하고 하위 두 번째 LSB는 Level 5의 변형 블록을 검출하기 위한 정보를 삽입한다.

따라서 원 영상에서 하위 두 개의 LSB는 0으로 초기화 한 후 Fig. 1과 같이 여러 Level로 분할하여 다음의 과정으로 각 Level 블록들에 대한 워터마크를 생성한다.

① Level 0 블록을 해쉬 함수의 입력으로 사용하여 128비트의 해쉬 코드를 생성한 후 생성된 해쉬 코드를 대칭키 암호로 암호화하여 128비트의 워터마크를 생성한다.

② Level 1의 블록들을 각각 해쉬 함수의 입력으로 사용하여 128비트의 해쉬 코드 4개를 생성하고, 생성된 해쉬 코드는 각각 대칭키 암호로 암호화하여 128비트 크기를 가지는 4개의 워터마크를 생성한다.

③ Level 2의 블록들은 16개의 블록으로 구성된다. Level 2의 블록들도 Level 1에서 워터마크를 생성한 방법과 동일하게 각 블록들을 해쉬 함수의 입력으로 사용하여 128비트의 해쉬 코드를 각각 생성하고, 생성된 해쉬 코드를 각각 대칭키 암호로 암호화하여 128비트 크기를 가지는 16개의 워터마크를 생성한다.

④ Level 3 블록은 총 64개의 블록으로 구성되고, Level 3의 블록들도 각각 해쉬 함수의 입력으로 사용하여 128비트의 해쉬 코드를 생성한 후, 생성된 해쉬 코드를 대칭키 암호로 암호화하여 128비트 크기를 가지는 64개의 워터마크를 생성한다.

⑤ Level 4 블록들은 256개로 구성되어 있어 ①~④의 방법과 동일하게 워터마크를 생성하게 되면 생성되는 워터마크의 크기가 너무 커지게 되어 Level 3 블록의 LSB에 모두 삽입할 수 없는 문제가 발생한다. 따라서 Level 4에서는 세로중복검사(LRC: Longitudinal Redundancy Check)를 위한 비트를 워터마크로 사용한다. Fig. 3에 나타낸 것과 같이 LRC 비트는 8×8 블록 상위 6비트들에 대한 패리티 비트로 각 블록별로 6비트씩 생성되어 상위 Level인 Level 3 블록 하나에는 총 24 개의 LRC 비트가 생성되어 워터마크를 삽입하는데 문제점이 발생하지 않는다.

	2^7	2^6	2^5	2^4	2^3	2^2
Pixel 1	1	0	1	1	0	1
Pixel 2	1	0	1	1	1	1
Pixel 3	0	1	1	0	0	1
⋮				⋮		
Pixel 64	1	1	0	1	1	0
Parity bit	1	0	1	1	0	1

Fig. 3. LRC bit generation

⑥ Level 5에서도 Level 4와 마찬가지로 많은 블록들이 생성되기 때문에 최소의 크기를 가지는 워터마크를 생성해야 한다. 따라서 Level 5에서는 Fig. 4에 나타낸 것과 같이 각 화소의 상위 6개 비트들에 대한 패리티 비트를 워터마크로 생성하여 각 화소의 하위 두 번째 LSB에 삽입한다. 따라서 패리티 비트를 검사하여 변형 여부를 판단하고 4×4 화소 단위로 변형 블록을 검출한다.

	2^7	2^6	2^5	2^4	2^3	2^2	Parity bit
Pixel 1	1	0	1	1	0	1	0
Pixel 2	1	0	1	1	1	1	1
Pixel 3	0	1	1	0	0	1	1
				⋮			⋮
Pixel 16	1	1	0	1	1	0	0

Fig. 4. The parity bit generation

4. 워터마크 삽입 및 검출

4.1 워터마크 삽입

Level 3 하나의 블록과 연관이 있는 Level 0 부터 Level 4까지의 블록들에서 생성된 워터마크는 Level 3 블록의 LSB 삽입된다. 그러나 Level 3에서 블록 하나의 화소는 총 256개로 구성되므로 각 블록의 LSB에 삽입할 수 있는 최대 비트 수는 256개로 제한된다. 따라서 Level 0 부터 Level 4까지의 워터마크는 총 536개로 구성되어 256개의 LSB 공간에는 536 비트를 삽입할 수 없다. 이러한 문제를 해결하기 위하여 Level 0 ~ Level 2에 포함되는 블록들에 대한 워터마크는 다음과 같이 분할하여 Level 3 블록에 삽입한다. 그리고 Level 3과 Level 4의 워터마크는 각각 128개의 비트와 24개의 비트로 구성되므로 총 152개의 비트 전체는 그대로 LSB에 삽입한다.

① Level 0은 하나의 블록으로 구성되고 Level 3의 64개 블록을 포함하고 있다. 따라서 Level 0 블록에 대한 128 비트 크기의 워터마크는 Level 3의 64개의 블록에 나누어 삽입하여도 문제가 발생하지 않으므로 각 블록별로 2비트씩 삽입한다.

② Level 1 하나의 블록은 Level 3의 16개 블록을 포함하고 있으므로 Level 1 하나의 블록 128 비트의 워터마크는 해당하는 하위 Level 3의 16개 블록에 8비트씩 나누어 삽입한다.

③ Level 2 하나의 블록은 4개의 Level 3 블록을 포함하고 있으므로 128 비트의 워터마크를 4개의 블록에 32비트씩 나누어 삽입한다.

이와 같은 과정으로 Level 3 블록 LSB에 삽입되는 워터마크는 다음과 같이 구성된다.

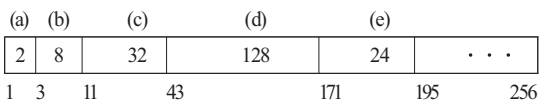

Fig. 5. LSB structure of the level 3 block

4.2 워터마크 검출

Level 3 블록에 삽입된 워터마크의 추출 과정은 워터마크의 삽입 과정과 반대로 구성된다. 즉, 워터마크가 삽입된 영상을 Fig. 1과 같이 여러 Level로 나누고 Level 3 블록의 하위 두개의 LSB에서 워터마크를 추출한 후 0으로 초기화 한다. Fig. 5의 (a)에 나타낸 것과 같이 Level 3 전체 64개 블록의 LSB에서 2비트씩 추출하여 복호화한 후 초기화된 전체 영상을 워터마크 생성에 사용된 해쉬 함수의 입력으로 사용하여 생성된 해쉬 코드를 복호화 된 해쉬 코드와 비교한다. 만약, 비교한 값이 동일하면 워터마크가 삽입된 영상에 변형이 발생하지 않은 것이므로 검사를 종료한다. 그러나 서로 다른 값으로 나타나면 워터마크가 삽입된 영상에 변형이 발생한 것이므로 Level 1의 4개의 블록 중 하나의 블록에 대한 해쉬 함수를 구한 후 Fig. 5의 (b)에 나타낸 것과 같이 해당 블록에 포함되는 하위 Level 3의 각 블록에서 8비트씩 추출하여 추출된 128 비트를 복호화 하여 초기화된 Level 1의 해당 블록의 해쉬 코드와 비교한다. 비교 결과 동일한 값이면 Level 1의 해당 블록은 변형이 발생하지 않은 것이므로 그 블록에 포함되는 하위 Level의 블록들에 대한 검사는 중단하고, Level 1의 다른 블록을 검사한다. 만약 계산된 해쉬 코드와 추출되어 복호화 된 해쉬 코드가 서로 다를 경우에만 해당 블록에 포함되는 Level 2 블록들을 Level 1 블록의 검사 방법과 동일한 방법으로 비교하여 변형이 발생된 블록에 대해서만 하위 Level의 블록들을 검사한다. 즉, Fig. 5의 (a) 비트들은 전체 영상에 대한 변형 여부를 검사하기 위한 워터마크의 일부분이며, (b)는 Level 1 블록들에 대한 변형 여부를 검사하기 위한 워터마크의 일부분이다.

Level 2의 변형 여부를 검사하기 위한 워터마크의 일부분은 (c)에 나타내었으며, (d)와 (e)는 각각 Level 3와 Level 4를 검사하기 위한 워터마크이다. 24비트로 구성된 (e)를 검사하면 Level 4의 8×8 블록들에 대한 변형 여부를 확인할 수 있으며, 변형이 발생된 블록에 대해서는 Level 3 블록의 하위 두 번째 LSB에서 추출된 패리티 비트와 Level 5에서 계산된 패리티 비트를 비교하여

변형이 발생된 화소를 4×4 블록 단위로 검출한다.

5. 결론

본 논문에서는 영상을 여러 Level로 분할하여 각 Level 블록과 연관된 하위 Level 블록들에 대한 워터마크를 16×16 블록의 하위 두 개의 LSB에 삽입하여 변형이 발생된 부분을 4×4 블록 단위로 검출할 수 있는 방법을 제안하였다. 기존에 제안된 방법에서는 변형이 발생된 부분을 16×16 블록 단위로 검출하기 때문에 한 개의 화소만 변형되어도 16×16 블록 단위로 검출되는 단점을 가지고 있었으나 본 논문에서는 이러한 문제를 개선하기 위한 방법을 제안하였다. 제안 방법에서 공개키 암호를 사용할 경우 암호문의 길이가 너무 커져서 워터마크를 16×16 블록의 LSB에 삽입하지 못하는 문제가 발생할 수 있어 대칭키 암호를 사용한 방법을 제안하였다. 제안 방법은 공간영역에서 하위 두 개의 비트에 워터마크를 삽입하기 때문에 시각적으로 화질 저하를 발견하기 어려우나, 하위 두 개의 LSB를 제거하는 공격과 정당한 사용자에 의한 변형이 발생할 수 있기 때문에 이러한 문제를 해결할 수 있는 방안에 대한 연구가 필요할 것으로 판단된다.

References

[1] P. W. Wong, “A Public Key Watermark for Image Verification and Authentication,” *Proc. of IEEE Conference on Image Processing*, pp. 425-429, 1998.

[2] P. L. Lin, C. K. Hsieh, P. W. Huang, “A hierarchical digital watermarking method for image tamper detection and recovery” Pattern Recognition 38, pp. 2519-2529, 2005.
DOI: http://dx.doi.org/10.1016/j.patcog.2005.02.007

[3] G. Kaur, K. Kaur, "Image Watermarking using LSB," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 4, pp. 858-861, April 2013.

[4] C. M. Wu, Y. S. Shin, "A Simple Image Tamper Detection and Recovery Based on Fragile Watermark with One Parity Section and Two Restoration Sections," Optics and Photonics Journal 3, pp. 103-107, 2013.
DOI: http://dx.doi.org/10.4236/opj.2013.32B026

[5] H. Nyeem, W. Boles and C. Boyd, "Counterfeiting Attacks on Block-Wise Dependent Fragile Watermarking Schemes" *Proc. of the 6th International Conference on Security of Information and Networks*, ACM Press and Digital Library, 2013.
DOI: http://dx.doi.org/10.1145/2523514.2523530

[6] C. I. Woo, S. D. Lee, “Digital Watermarking for Image Tamper Detection using Block-Wise Technique” International Journal of Smart Home, vol. 7, no. 5, pp. 115-124, 2013.
DOI: http://dx.doi.org/10.14257/ijsh.2013.7.5.12

[7] D. H. Won, *Modern Cryptology*, pp. 99-270, Green Press, 2006.

우 찬 일(Chan-Il Woo) [종신회원]

- 1995년 2월 : 단국대학교 대학원 전자공학과 (공학석사)
- 2003년 2월 : 단국대학교 대학원 전자공학과 (공학박사)
- 1995년 11월 ~ 1997년 2월 : LG 이노텍(주) 연구원
- 2004년 3월 ~ 현재 : 서일대학교 정보통신과 교수

<관심분야>
정보보호, 암호 프로토콜, 디지털워터마킹

이 승 대(Seung-Dae Lee) [정회원]

- 1992년 2월 : 단국대학교 대학원 전자공학과 (공학석사)
- 1999년 8월 : 단국대학교 대학원 전자공학과 (공학박사)
- 1995년 4월 ~ 현재 : 남서울대학교 전자공학과 교수

<관심분야>
정보통신, 유무선통신, 정보보호