Journal of the Korea Academia-Industrial cooperation Society
Vol. 21, No. 11 pp. 878-883, 2020

https://doi.org/10.5762/KAIS.2020.21.11.878
ISSN 1975-4701 / eISSN 2288-4688

# RSA와 해시 함수 기반 이미지 무결성 검증에 관한 연구

우찬일[1], 구은희[2*]
[1]서일대학교 정보통신공학과, [2]아주대학교 다산학부대학

# A Study on Image Integrity Verification Based on RSA and Hash Function

Chan-Il Woo[1], Eun-Hee Goo[2*]
[1]Department of Information and Communication Engineering, Seoil University,
[2]Dasan University College, Ajou University

**요 약** 데이터에 대한 불법적인 조작을 방지하기 위해 사용되는 암호 알고리즘은 공개키 암호와 대칭키 암호로 나누어진다. 공개키 암호는 대칭키 암호에 비하여 암호화와 복호화에 많은 시간이 소모되는 단점이 있으나 암호화와 복호화에 서로 다른 키를 사용하기 때문에 대칭키 암호에 비하여 키 관리와 배송이 쉬운 장점이 있다. 그리고 다양한 크기의 데이터를 입력으로 사용하여 항상 고정된 크기의 출력을 생성하는 해시 함수는 디지털 콘텐츠의 무결성 검증을 위해 매우 효과적으로 사용되고 있다. 본 논문에서는 디지털 영상의 변형 여부와 변형 위치를 검출하기 위해 RSA 공개키 암호와 해시 함수를 이용한 방법을 제안한다. 제안 방법에서는 전체 영상을 64×64 크기를 갖는 여러 개의 블록으로 나눈 후 각 블록에 대한 워터마크를 생성하여 해당 블록의 변형 여부를 확인한다. 그리고 블록 내에서 변형이 발생 된 화소는 4×4 크기를 갖는 여러 개의 서브 블록으로 분할하여 각각의 서브 블록에 대한 워터마크를 생성하여 검출한다. 제안 방법의 안전성은 암호 알고리즘과 해시 함수의 안전성에 의존한다.

**Abstract** Cryptographic algorithms are used to prevent the illegal manipulation of data. They are divided into public-key cryptosystems and symmetric-key cryptosystems. Public-key cryptosystems require considerable time for encryption and decryption compared to symmetric-key cryptosystem. On the other hand, key management, and delivery are easier for public-key cryptosystems than symmetric-key cryptosystems because different keys are used for encryption and decryption. Furthermore, hash functions are being used very effectively to verify the integrity of the digital content, as they always generate output with a fixed size using the data of various sizes as input. This paper proposes a method using RSA public-key cryptography and a hash function to determine if a digital image is deformed or not and to detect the manipulated location. In the proposed method, the entire image is divided into several blocks, 64×64 in size. The watermark is then allocated to each block to verify the deformation of the data. When deformation occurs, the manipulated pixel will be divided into smaller 4×4 sub-blocks, and each block will have a watermark to detect the location. The safety of the proposed method depends on the security of the cryptographic algorithm and the hash function.

**Keywords :** Cryptography, Steganography, LSB, RSA, Hash Function

본 논문은 2020년도 서일대학교 학술연구비에 의해 연구되었음
*Corresponding Author : Eun-Hee Goo(Ajou Univ.)
email: ehgoo@ajou.ac.kr
Received July 14, 2020 Revised July 30 2020
Accepted November 6, 2020 Published November 30, 2020

## 1. 서론

디지털 콘텐츠에 대한 저작권을 증명하거나 인증이나 무결성을 검증하기 위한 용도로 개발된 디지털 워터마킹은 주파수 영역과 공간 영역에서 다양한 방법들이 제안되고 있으며 디지털 콘텐츠에 대한 거래를 추적하는 용도로도 활용되고 있다. 저작권을 보호하기 위한 워터마킹은 저작권을 증명하기 위해 삽입된 정보는 다양한 공격에도 저작권 정보가 제거되지 않아야 저작권을 증명할 수 있기 때문에 워터마크의 강인성이 매우 중요하다. 따라서 저작권을 보호하기 위한 기술에서는 영상의 경우 필터링이나 왜곡과 같은 공격에도 삽입된 워터마크는 추출될 수 있어야 한다[1-2]. 저작권 보호가 아닌 워터마킹의 다른 응용 분야로는 인증이나 무결성을 증명하기 위한 기술이 있다[3-10]. 이 기술은 영상을 임의로 조작할 경우 영상의 조작 여부와 조작 위치를 확인할 수 있는 기술로써 워터마크가 삽입된 영상에 공격이 가해 질 경우 저작권 보호 기술과는 반대로 삽입된 정보는 쉽게 제거될 수 있어야 한다.

따라서 저작권 보호 기술에서는 워터마크가 삽입된 콘텐츠에 다양한 공격이 인가될 경우 저작권을 증명하기 위해 삽입된 정보는 지워지지 않고 추출되어야 하고, 인증이나 무결성을 증명하기 위한 기술에서는 공격이 인가될 경우 공격 여부를 확인하기 위해 삽입된 워터마크가 쉽게 제거될 수 있는 특성을 가져야 한다. 또한 워터마크는 공간 영역과 주파수 영역에서 삽입할 수 있으며, 주파수 영역에서 삽입할 경우 공간 영역에서 삽입하는 경우보다 강인성이 향상될 수 있다. 따라서 대부분의 저작권 보호 기술에서는 주파수 영역에서 워터마크를 삽입하고 있다. 그러나 인증과 무결성을 증명하기 위한 기술에서는 삽입된 워터마크가 쉽게 제거되어야 하는 특성 때문에 공간 영역에서 워터마크를 삽입하는 방법들이 다수 제안되고 있으며, 공간 영역 워터마킹 기술 중 워터마크를 임의로 변경하는 것을 방지하기 위해 암호학적으로 안전한 해시 함수와 공개키 암호 알고리즘을 사용하는 방법들도 제안되고 있다[1-4].

본 논문에서는 영상의 무결성을 검증하기 위한 방법을 RSA 공개키 암호와 암호학적으로 안전한 해시 함수를 기반으로 제안한다. 제안 방법에서는 원 영상을 64×64 크기의 블록으로 나누어 전체 영상의 변형 여부를 검사하고 만약, 변형이 발생 된 블록이 있을 경우 다시 4×4 크기의 블록으로 분할하여 변형된 화소를 검출한다. 제안 방법의 안전성은 공개키 암호와 해시 함수의 안전성에 기반을 두고 있으며 공개키를 이용하여 영상의 변형 여부를 확인할 수 있다.

## 2. 관련 연구

### 2.1 워터마킹 기술 분석

워터마크의 강인성 향상을 위해서는 Fig. 1과 같이 DCT(discrete cosine transform), FFT(fast fourier transform) 등을 이용하여 원 영상을 주파수 영역으로 변환한 후 워터마크를 삽입하는 방법들이 제안되고 있다[2].

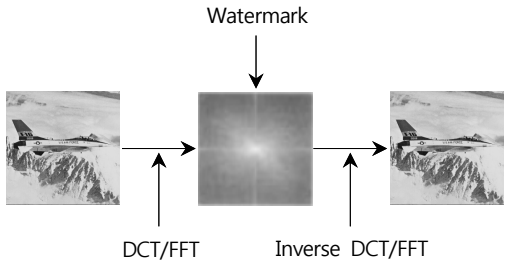


Fig. 1. Watermark insertion process(frequency domain)

그러나 공간 영역에서 수행되는 워터마킹은 워터마크의 강인성 보장이 어려워 인증이나 무결성을 검증하기 위한 용도로 주로 사용되고 있으며, Fig. 2와 같이 영상을 작은 크기의 여러 블록으로 나눈 후 공개키 암호를 사용하여 워터마크를 삽입하는 방법이 제안되었다[3].

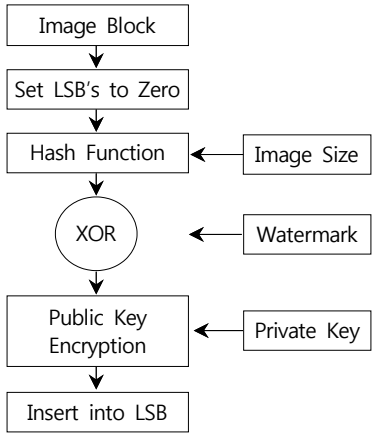


Fig. 2. Watermark insertion process(spatial domain)

### 2.2 무결성 검증 기술

디지털 콘텐츠에 대한 무결성을 검증하는 방법은 암호학적으로 안전한 일방향 해쉬 함수(one way hash function)를 사용하는 것이 가장 효과적이며, 해시 함수는 디지털 서명, 메시지 인증 코드 등 정보보호 분야에서 널리 사용되고 있다. 해시 함수를 이용하여 무결성을 검증하는 과정은 Fig. 3과 같다.

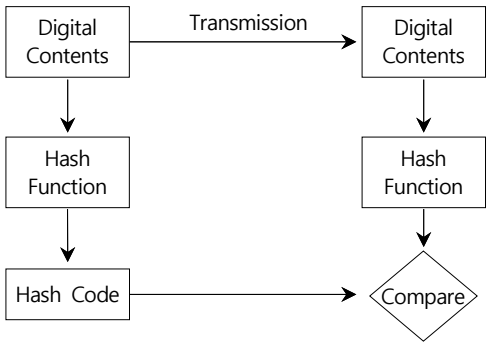


Fig. 3. Integrity verification process

### 2.3 공개키 암호

암호 알고리즘은 비밀키(대칭키) 암호와 공개키(비대칭키) 암호로 나누어진다. 비밀키 암호는 암, 복호화에 사용되는 키가 동일하고 암, 복호화 속도가 빠른 장점이 있으나 키 관리와 키 배송이 어려운 단점이 있다. 그러나 공개키 암호는 수학적 연산으로 인하여 암, 복호화 속도는 느리지만 암, 복호화에 서로 다른 키를 사용하기 때문에 키 관리와 배송이 쉬운 장점이 있다.

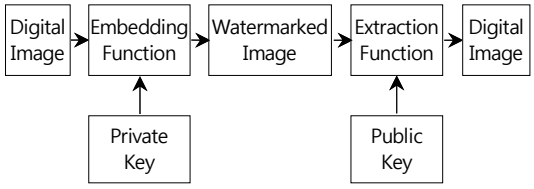


Fig. 4. Watermark insert and extraction process with public key cryptosystem

Fig. 4는 공개키 암호를 이용하여 워터마크를 삽입하고 추출하는 과정을 나타내고 있다. 일반적으로 공개키 암호시스템에서 데이터를 암호화하여 전송하기 위해서는 수신자의 공개키를 이용하여 데이터를 암호화하고 수신자는 개인키를 이용하여 암호화된 메시지를 복호화한다. 공개키 암호 시스템에서는 공개키를 쉽게 얻을 수 있기 때문에 암호화는 누구나 수행할 수 있지만 복호화는 개인키를 소지한 사람만 수행할 수 있다. 그러나 암호화와 복호화에 공개키와 개인키를 반대로 사용하게 되면 암호화는 개인키를 가진 사람만이 수행할 수 있고 공개키를 알고 있는 사람이면 누구나 복호화를 수행할 수 있게 된다. 따라서 Fig. 4와 같이 워터마크 삽입을 위해 개인키를 사용하고 워터마크 검출에 공개키를 사용할 경우 개인키를 가지고 있는 사람만이 워터마크를 삽입할 수 있고 워터마크의 검증은 공개키를 이용하여 누구나 가능하기 때문에 인증이나 무결성을 검증하기 위한 워터마킹에 매우 효과적으로 사용할 수 있다.

## 3. 워터마크 생성

### 3.1 워터마크 삽입을 위한 영상 구조

제안 방법에서는 128×128 화소 크기의 영상을 기준으로 Fig. 5와 같이 64×64 크기의 영상으로 분할하고 64×64 영상은 다시 4×4 크기의 영상으로 분할하여 워터마크를 삽입한다. 제안 방법에서는 워터마크가 삽입된 전체 영상에 대한 변형 여부는 64×64 블록 단위로 검사하고, 변형이 발생 된 64×64 블록에 대해서는 다시 4×4 블록으로 나누어 변형 화소를 검출한다. 따라서 64×64 크기의 영상 블록은 변형이 발생 된 영역을 검출하기 위한 용도로 사용되고 블록 내에서 변형이 발생 된 화소는 4×4 블록 단위로 검출한다.

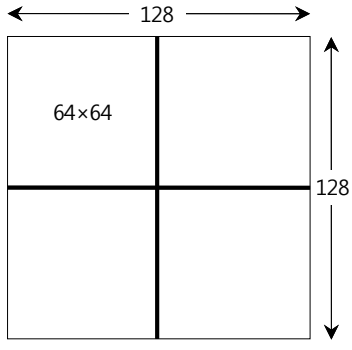


Fig. 5. Image segmentation

Fig. 5에서 64×64 크기의 각 블록들은 블록별로 서로 다른 워터마크를 생성한다. 64×64 블록 단위로 생성되는 워터마크는 해당 블록에 대한 변형 여부를 검사하기 위한 워터마크로 사용되고 만약, 변형이 발생 되었을

경우 변형이 발생 된 위치는 화소 단위로 찾는 것이 매우 어렵기 때문에 최소 크기를 갖는 블록 단위로 검사를 진행한다. 따라서 변형이 발생 된 블록에 대해서는 4×4 블록으로 분할하여 변형이 발생 된 화소 영역을 검출한다.

## 3.2 블록별 워터마크 생성

### 3.2.1 64×64 블록 변형 검출을 위한 워터마크

본 논문에서는 일방향 해시 함수와 RSA 공개키 암호 알고리즘을 이용하여 64×64 블록에 대한 워터마크를 생성한 후 블록 내 화소의 LSB에 워터마크를 삽입하며, 워터마크의 생성 과정은 다음과 같다.

① 전체 영상의 LSB를 '0'으로 초기화하고 64×64 크기를 갖는 여러 블록으로 분할한다.
② 64×64 블록을 각각 해시 함수의 입력으로 사용하여 해시 코드를 생성한다.
③ 해시 코드를 RSA 공개키 암호의 개인키로 암호화한다. RSA 공개키 암호 알고리즘에서 N의 크기는 2,048을 적용하여 최대 2,048 비트의 암호문을 생성한 후 워터마크로 사용한다.

64×64 블록으로 생성된 워터마크는 해당 블록에 대한 변형 여부를 확인하기 위해 사용되며 블록 내에서 하나의 화소만 변형되더라도 64×64 블록 단위로 변형 여부를 검출하기 때문에 변형이 발생 된 화소는 보다 작은 블록으로 나누어 검출하는 것이 효과적이다. 따라서 64×64 블록은 다시 256개의 4×4 블록으로 나누어 4×4 블록의 LSB에 워터마크를 삽입한다. 64×64 블록은 총 4,096개의 화소로 구성되고 워터마크의 최대 크기는 2,048 비트의 크기를 가지기 때문에 64×64 블록에 대한 변형 검출을 위해 삽입되는 워터마크는 블록 내에서 1/2 크기에 해당하는 2,048개의 화소만 필요하다. 따라서 본 논문에서는 64×64 블록을 4×4 블록으로 나눈 후 16개의 LSB 중 1/2에 해당하는 상위 8개 화소의 LSB에 64×64 블록에 대한 변형 검출을 위한 워터마크를 삽입한다.

### 3.2.2 변형 화소 검출을 위한 워터마크

변형이 발생 된 64×64 블록은 4×4 블록으로 분할하여 변형이 발생 된 화소를 검출하기 때문에 4×4 블록 검사를 위한 워터마크의 생성은 다음과 같다.

① 초기화된 64×64 블록을 4×4 블록으로 나눈다.
② 4×4 블록 내 화소들에 대한 평균을 구하고 소수점 이하는 제거한다.
③ 4×4 블록 내의 각 화소값($P_i$)을 소수점을 제거한 화소 평균값($Block_{average}$)과 식 (1)을 이용하여 계산한 후 Fig. 6과 같이 16 비트를 생성한다.

$$v = \begin{cases} 1: & if\ P_i \geq Block_{average} \\ 0: & otherwise \end{cases} \quad (1)$$

| | | | |
|---|---|---|---|
| 124 | 124 | 126 | 122 |
| 118 | 120 | 122 | 120 |
| 118 | 120 | 120 | 122 |
| 120 | 118 | 118 | 120 |

⇨

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |

Fig. 6. Process of generating 16-bits

④ 초기화된 4×4 블록과 개인키 그리고 Fig. 6의 16 비트를 SHA-256 해시 함수의 입력으로 사용하여 256 비트의 출력을 생성한다.
⑤ 해시 코드 256 비트는 8 비트씩 분할한 후 32개의 8 비트 블록을 Fig. 7과 같이 XOR 연산을 수행하여 W(8비트)를 구한다.

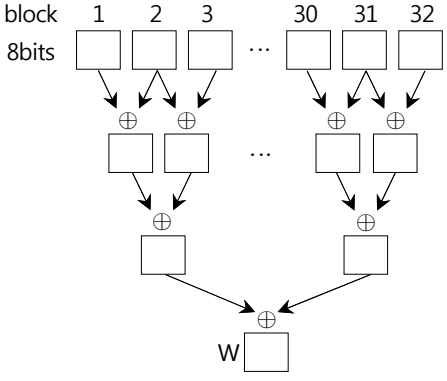


Fig. 7. Process of generating 8-bits for watermark

⑥ W는 소수점을 제거한 화소 평균값과 XOR 연산을 수행하여 8 비트의 워터마크를 생성한다.

$$mark = W \oplus \text{화소 평균값} \quad (2)$$

## 4. 워터마크 삽입 및 변형 검출

### 4.1 워터마크 삽입

본 논문에는 64×64 블록에 대한 변형 여부를 검사하기 위한 워터마크와 4×4 블록에 대한 변형 여부를 검사하기 위한 워터마크를 각각 생성하여 삽입한다. 64×64 블록을 검사하기 위한 워터마크는 전체 영상에 대한 변형 여부를 확인하기 위해 사용하고 만약, 변형이 발생 되었을 경우 해당 블록을 다시 4×4 블록으로 나누어 검사를 수행하여 변형이 발생 된 화소 영역을 검출한다.

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Fig. 8. 4×4 block

Fig. 8은 4×4 블록을 구성하고 있는 16개의 화소를 나타내고 있다. 4×4 블록 내에서 1~8번째까지 화소의 LSB에는 64×64 블록에 대한 변형 여부를 확인하기 위한 워터마크를 삽입하고 9~16번째까지의 화소에는 4×4 블록에 대한 변형 여부를 확인하기 위한 워터마크를 삽입한다. 64×64 블록은 총 256개의 4×4 블록으로 구성되고 64×64 블록에 대한 변형을 검출하기 위한 워터마크의 최대 크기는 2,048 비트를 가진다. 따라서 4×4 블록의 상위 8개의 LSB에 워터마크를 삽입할 경우 총 2,048개의 비트 삽입이 가능하므로 RSA 공개키 암호로 암호화된 64×64 블록의 변형 검출을 위한 워터마크를 삽입하는데 충분한 공간을 확보할 수 있다. 그리고 4×4 블록에서 하위 8개의 LSB에는 식 (2)에서 구한 워터마크가 삽입되어 변형이 발생 된 화소를 4×4 블록 단위로 검출하기 위해 사용한다.

### 4.2 변형 검출

워터마크가 삽입된 영상에 대한 변형 검출은 다음과 같은 과정으로 수행된다.

① 워터마크가 삽입된 영상을 64×64 블록 단위로 분할한다.
② 64×64 블록을 4×4 블록으로 나누고 4×4 블록에서 상위 8개 화소의 LSB와 하위 8개 화소의 LSB에서 각각 8비트씩 추출하고 LSB를 '0'으로 초기화한다. 상위 8개 화소에서 추출된 8 비트는 64×64 블록 내의 모든 4×4 블록들의 상위 8개 화소에서 추출한 8 비트와 합하여 RSA의 공개키로 복호화한다.
③ 초기화된 64×64 블록을 해시 함수의 입력으로 사용하여 생성된 해시 코드를 복호화된 해시 코드와 비교한다. 이러한 과정은 전체 블록을 대상으로 수행하며, 비교 결과 같은 값이면 변형이 발생 되지 않은 것에 해당하므로 검사를 종료한다. 그러나 추출된 해시 코드와 생성된 해시 코드가 서로 다를 경우 해당 블록을 여러개의 4×4 블록으로 나눈다.
④ 워터마크 생성 과정과 동일한 과정으로 4×4 블록에 대한 워터마크를 생성한 후 추출된 워터마크와 비교하여 서로 다른 값이 나타날 경우 변형이 발생된 블록으로 판단한다.

본 논문에서 제안하는 방법은 2종류의 워터마크를 사용하기 때문에 전체 영상의 변형 여부는 공개키를 이용하여 누구나 확인이 가능할 수 있으나 변형이 발생된 화소의 위치는 개인키를 가지고 있는 사람만이 검출할 수 있다.

## 5. 결론

디지털 워터마킹은 디지털 콘텐츠에 대한 저작권 보호와 인증이나 무결성을 검증하기 위한 용도로 개발되었다. 본 논문에서는 인증과 무결성을 검증하기 위한 워터마킹을 공개키 암호와 해시 함수를 기반으로 제안하였다. 제안 방법에서는 전체 영상을 64×64 블록으로 나누어 변형 유, 무를 확인하고 변형이 발생 된 블록은 4×4 크기를 갖는 작은 블록들로 나누어 변형이 발생 된 화소 영역을 검출한다. 이와 같은 과정을 수행하면 변형이 발생 된 64×64 블록에 대해서만 작은 블록 단위로 검사를 수행하기 때문에 전체 영상을 검사하는 시간을 줄일 수 있고 워터마크는 해시 코드를 개인키로 암호화하여 생성하기 때문에 임의로 변경하는 것이 불가능하다. 그리고 4×4 블록에 대한 워터마크는 RSA 공개키 암호를 이용하여 생성할 경우 최대 2,048 비트 크기를 갖는 워터마크가 생성될 수 있기 때문에 생성되는 워터마크를 삽입하기 위한 공간이 부족하여 암호화를 수행할 수 없다. 따라서 워터마크를 삽입하기 위한 공간 문제를 해결하고 삽입된

워터마크를 임의로 변경하는 것을 방지하기 위해 워터마크를 생성한 사람만이 소지하고 있는 개인키를 이용하여 워터마크를 생성한다. 이와 같은 과정으로 워터마크를 생성하여 삽입하면 전체 영상에 대한 변형 여부는 누구나 확인할 수 있지만 변형이 발생 된 화소 영역은 워터마크를 생성한 사람만이 확인할 수 있다. 제안 방법은 암호학적으로 안전한 공개키 암호와 해시 함수의 안전성에 의존하고 있어 워터마크를 조작하기 위한 다양한 공격으로부터 안전성이 보장될 수 있다.

## References


[1] H. S. Kim, Digital Watermarking, p. 546, Green Publishing, pp. 17-19, 2005.

[2] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," IEEE Transactions on image processing, Vol. 6, No. 12, pp. 1673-1687, 1997.

[3] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," in Proc. of IEEE Conf. on Image Processing, pp. 425-429, 1998.
DOI: https://dx.doi.org/10.1109/ICIP.1998.723526

[4] R. Halder, S. Sengupta, S. Ghosh, D. Kundu, "A Secure Image Steganography Based on RSA Algorithm and Hash-LSB Technique," IOSR Journal of Computer Engineering(IOSR-JCE), Vol. 18, Issue 1, Ver. Ⅳ, pp. 39-43, 2016.

[5] Yoo, Heung-Ryol, Son, Yung-Deug, "Fragile Watermark System using Quantization and DC Coefficients", Journal of IKEEE, Vol. 22, No. 3, pp. 774-779, 2018.
DOI: https://doi.org/10.7471/ikeee.2018.22.3.774

[6] P. MeenakshiDevi, M. Venkatesan and K. Duraiswamy, "A Fragile Watermarking Scheme for Image Authentication with Tamper Localization Using Integer Wavelet Transform", Journal of Computer Science, Vol. 5, No. 11, pp. 831-837, 2009.
DOI: https://doi.org/10.3844/jcssp.2009.831.837

[7] A.Kannammal, S.Subha Rani, "Authentication of DICOM Medical Images using Multiple fragile watermarking Techniques in Wavelet Transform Domain", International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, pp. 181-189, 2011.

[8] S. Dadkhah, A. Abd Manaf and S. Sadeghi, "Efficient Digital Image Authentication and Tamper Localization Technique Using 3Lsb Watermarking", International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, pp. 1-8, 2012.

[9] Heng Zhang, Chengyou Wang, and Xiao Zhou, "Fragile Watermarking Based on LBP for Blind Tamper Detection in Images", Journal of Information Processing Systems, Vol. 13, No. 2, pp. 385-399, 2017.
DOI: https://doi.org/10.3745/JIPS.03.0070

[10] P. Rahmati, A. Adler, T. Tran, " Watermarking in E-commerce", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 6, pp. 256-265, 2013
DOI: https://doi.org/10.14569/ijacsa.2013.040634



**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 정보통신공학과 교수

〈관심분야〉
정보보호, 암호 프로토콜, 디지털워터마킹, 소프트웨어공학

**구 은 희(Eun-Hee Goo)** [정회원]

- 2004년 8월 : 단국대학교 대학원 전자컴퓨터공학과 (공학석사)
- 2009년 8월 : 단국대학교 대학원 전자컴퓨터공학과 (공학박사)
- 2013년 3월 ~ 2014년 9월 : ㈜도넛시스템LSI 책임연구원

•2014년 10월 ~ 2016년 8월 : ㈜이너트론 수석연구원
•2016년 9월 ~ 현재 : 아주대학교 다산학부대학 교수

〈관심분야〉
정보보호, 암호 알고리즘, 서비스로서의 보안(ASCaaS), 소프트웨어공학