논문 2024-4-11 http://dx.doi.org/10.29056/jsav.2024.12.11

고속 푸리에 변환의 버터플라이 알고리즘에서 데이터 암호화 마스크 적용 연구

송재훈*, 이연호**†

Study on Application of Encryption Mask in Butterfly Algorithm of fast Fourier Transform

Jae-Hun Song*, Yeon-Ho Lee**†

요 약

IT산업의 급속한 발전으로 디지털 기술에 대한 의존도가 높아지고 이전과는 차원이 다른 방대한 데이터를 처리하게 되었다. 특히 이러한 데이터는 민감한 개인정보 및 기술 정보 등을 포함함에 따라 해킹 및 복제를 방지하고 안전하게 보호하기 위한 데이터 보안기술이 필요하다. 본 논문에서는 고속 푸리에 변환의 버터플라이 알고리즘을 이용하여 이차원 이미지 데이터의 빠른 송/수신이 가능하게 하고, 기본 암호키를 구성하여 적용한다. 또한 데이터 보안성 향상을 위한 위상변조 하다마드 마스크, 데이터 셔플링 비트 시프트 마스크 등을 개발하여 데이터 암호화에 적용하고 최적의 암호화 마스크 제작 기법을 제안한다. 모든 암호키는 간단한 정수로 구성된다. 원본 및 암호화, 복원된 데이터와 상관계수 비교를 통해 제안된 암호화 마스크의 성능을 분석해 보았다. 다양한 시뮬레이션을 통해 제안된 암호화 마스크들이 정보보안 및 암호화 기술에 사용될 수 있음을 확인해본다.

Abstract

The rapid development of the IT industry has led to an increasing dependence on digital technologies, resulting in the need to process vast amounts of data on a scale previously unseen. In particular, since such data often includes sensitive personal and technical information, there is a growing demand for data security technologies to prevent hacking, replication, and ensure safe protection. In this paper, we propose an optimal encryption mask design method by leveraging the butterfly algorithm of the Fast Fourier Transform (FFT) to enable fast transmission and reception of two-dimensional image data. For data security, we develop phase-modulated Hadamard masks and bit-shifted masks. Additionally, we analyze the performance of the proposed encryption masks by calculating the correlation coefficient(CC) between the original, encrypted, and restored data.

한글키워드 : 정보보안, 암호화, 고속푸리에변환, 버터플라이, 하다마드, 비트시프트

keywords : Information Security, Encryption, Fast Fourier Transform, Butterfly, Hadamard, Bit-Shift

* 성균관대학교 IT융합연구원
** 성균관대학교 전자전기공학부
† 교신저자: 이연호(email: pfyeonlee@skku.edu)
접수일자: 2024.11.04. 심사완료: 2024.12.06.
게재확정: 2024.12.20.

1. 서 론

최근 광학장비, 인공지능(AI), 양자(Quantum)

기술 등의 발전은 해킹 및 복제를 당한 당사자도 구분이 힘들 정도로 정교한 기술 발전이 이뤄지고 있다. 이는 개인과 기업의 디지털 데이터의 의존도가 날로 높아지고 있는 상황에서 민감한 개인정보 및 기술 유출 또한 심각해지고 있는 상황이며, 그 결과 정보보안의 중요성과 차별화된 보안기술의 개발이 필요하게 되었다[1-3].

이러한 데이터 보안기술에서 홀로그램 등 3D 데이터의 암호화를 구현하기 위한 기술 또한 많이 연구되고 있다. 원본 이미지 픽셀 데이터가 가지는 값의 위상을 변조시키는 Random Phase Mask[4], 진폭이 일정한 두 개의 위상 암호화 마스크를 만들어 간섭 현상을 이용한 암호화[5], 이산 푸리에 변환 (DFT, Discrete Fourier Transform)에서 계산 식이 가지는 파라미터 값을 암호키로 사용하여 데이터를 암호화 시키는 방법[6], 원본 데이터에 임의의 노이즈 마스크들을 무수히 많이 적용하여 데이터를 암호화하고 앙상블 평균(Ensemble Average)을 이용하여 데이터를 복호화 시키는 방법[7] 등 여러 기술이 연구되고 있다.

기존에 소개된 암호화 방법들은 이미지 데이터가 암호화 될 때, 연산의 입력단 데이터 혹은 출력단 데이터에 2D암호키를 적용하므로 픽셀 수와 동일한 개수의 암호화 키가 필요했다. 이는 기억해야 할 암호키가 데이터가 커질수록 기하급수적으로 많아지고 정보의 위치가 노출되어 조작을 가능하게 한다. 또한 연산의 복잡도가 올라가 데이터 암호화에 시스템이 느려지는 등 여러 문제점이 발생하였다.

본 논문에서는 이를 해결하기 위해 고속푸리에변환 FFT(Fast Fourier Transform)의 버터플라이 알고리즘에서 원하는 경로에 접근하는 방법[8]에 하마마드(Hadamard), 비트 시프트(Bit-Shift)를 이용한 암호화 마스크 기법들을 소개하고 적용해 본다. 이를 통해 암호화 마스크의 기능과 성능을 시뮬레이션을 통해 분석해 본다.

2. 데이터 암호화 응용을 위한 기본 이론

2.1 고속 푸리에 변환에서 버터플라이 알고리즘

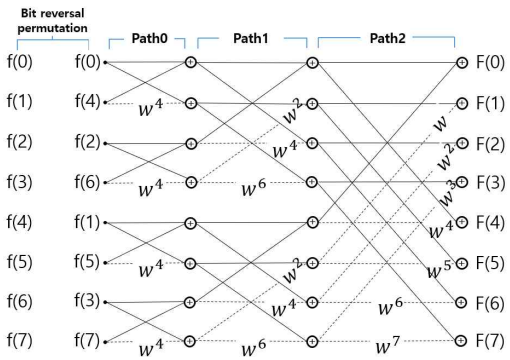

그림 1. M=8일 때 고속 푸리에 변환의 버터플라이 알고리즘(일차원)
Fig. 1. Butterfly algorithm of Fast Fourier Transform for M=8 (1D)

일반적인 이미지 데이터 $f(x, y)$의 이차원(2D) 푸리에 변환은 식 (1)과 같다.

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi(ux+vy)}dxdy \tag{1}$$

이러한 푸리에 변환은 디지털 영역에서 이산 푸리에 변환(DFT)으로 식 (2)와 같이 표현할 수 있다.

$$F(\alpha,\beta)=\sum_{a=0}^{M-1}\sum_{b=0}^{N-1} f(a,b)e^{-i2\pi(\frac{\alpha a}{M}+\frac{\beta b}{N})} \tag{2}$$

x축과 y축을 따라 샘플링 간격이 각각 X와 Y

일 때, 두 정수 a와 b는 다음 식(3)과 같이 정의된다.

$$a = \frac{x - x_{\min}}{X}, b = \frac{y - y_{\min}}{Y} \quad (3)$$

여기서 $x_{\min}$과 $y_{\min}$은 이차원 좌표에서 왼쪽 아래 모서리를 시작으로 하는 점의 좌표이고, M과 N은 각각 x 및 y축 방향의 샘플링 횟수이다. α와 β는 공간 정수(Spatial Integer)로 이를 통해 $\alpha/(NX)$와 $\beta/(NY)$로 각각 x 및 y축의 단위당 주기로 측정된 공간 주파수(Spatial Frequency)를 나타낸다. 일차원(1D) 데이터에 대한 DFT는 식(4)와 같이 표시되고, 복소수 부분은 트위들 팩터(Twiddle Factor) w를 이용하여 표현된다.

$$F[\alpha] = \sum_{a=0}^{M-1} f(a) w^{\alpha a} \quad (4)$$

트위들 팩터 w는 복소지수로 식 (5)로 정의되며,

$$w = \exp(-i2\pi/M) \quad (5)$$

실수부가 $\cos(2\pi/M)$, 허수부가 $\sin(2\pi/M)$으로 주어진다. 여기서 $i = \sqrt{-1}$이고, M은 픽셀 데이터 수를 의미한다. 이러한 DFT를 컴퓨터를 이용하여 계산하는 경우 버터플라이 알고리즘이 포함된 FFT가 널리 사용된다. 예를 들어 M이 8인 데이터에 대한 버터플라이 알고리즘에서는 그림 1과 같이 데이터 인덱스의 순열을 바꾸는 Bit Reversal Permutation 과정을 거친 후 연속적인 경로(Path)에서 더 작은 FFT 계산으로 분해 되며, 각 경로마다 대칭 및 주기성을 가진 w가 곱해져 연산이 수행된다.

일반적으로 픽셀 데이터 수가 M이면 FFT 경로의 개수는 $\log_2 M$으로 주어지고 버터플라이 연산이 수행되며 이를 통해 연산의 복잡성이 획기적으로 줄어들게 된다. M이 8인 경우 그림 1에서 보이는 것처럼 1D FFT는 3개의 경로를 거치면서 버터플라이 알고리즘 연산이 수행되는 것을 보여준다[9-11].

본 논문에서는 FFT의 특정 경로에서 픽셀 데이터를 암호화하므로, 몇 번째 경로인지 그 특정 경로의 번호 정보가 암호키로 사용되는 특징을 가진다.

2.2 하다마드(Hadamard) 개념

하다마드 코드는 일반적으로 n차 하다마드 행렬(H) 각 요소가 +1 또는 -1(0 또는 π)로 이루어진 $N \times N$ 정방행렬로 주어지며 그 행은 상호 직교성을 가진다.

$$H_{N\times N}^{T} H_{N\times N} = N I_{N\times N} \quad (6)$$

즉, 식(6)과 같으며 여기서 H^T는 행렬 H의 전치행렬이고, $I_{N\times N}$는 $N \times N$ 단위행렬이다.

Sylvester 구성법을 통해 다음 식 (7)와 같이 하다마드 행렬을 얻을 수 있다[12-15].

$$H_{1\times 1} = 1 \quad H_{2\times 2} = \begin{bmatrix} H_{1\times 1} & H_{1\times 1} \\ H_{1\times 1} & H_{1\times 1} \end{bmatrix}, \cdots,$$

$$H_{N\times N} = \begin{bmatrix} H_{(N/2)\times(N/2)} & H_{(N/2)\times(N/2)} \\ H_{(N/2)\times(N/2)} & H_{(N/2)\times(N/2)} \end{bmatrix} \quad (7)$$

여기서 $N = 2^m$이고 m은 정수이다. 하다마드 행렬은 다음과 같은 성질을 갖는다; (a)두 행 또는 두 열을 치환해도 식 (6)를 만족한다. (b) 행 또는 열을 부호 반전해도 식 (6)를 만족한다. 그리고 n차 Walsh-Hadamard 코드는 직교 조건을 만족해야 한다.

$$\sum_{i=1}^{N} S_i H_i^a H_i^b = NS\delta_{ab} \qquad (8)$$

여기서 식(8)의 $\delta_{ab} = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$ 는 Kronecker 델타 함수이고, S_i는 이미지 데이터의 정보이다. 하다마드 행렬의 이러한 특성을 이용하여, 우리는 다중 이미지를 중첩하거나 암호화하기 위한 직교 코드 행렬 집합을 생성한다. 간단히 하기 위해, 4bit 하다마드 행렬을 고려하면 다음 식 (9)와 같이 구성된다.

$$H_{4\times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad (9)$$

본 논문에서는 이미지의 각 픽셀에 하다마드 마스크를 적용해 본다. 이는 하다마드 코드가 가지는 간단함과 직교성의 특징을 이용해 위상변조 마스크로 사용할 수 있고 복잡한 키를 생성하지 않아도 고효율의 암호화 데이터를 획득할 수 있도록 한다.

2.3 비트 시프트(Bit-Shift) 개념

이미지의 픽셀 데이터에서 데이터 배열의 인덱스를 이진수로 표현하고 여기서 오른쪽(Right) 또는 왼쪽(Left) 비트 시프트를 수행함을 의미한다. 예를 들어, N=256인 데이터의 3번째 인덱스 즉, j=3를 이진수로 표현하면 0000 0011이고 여기에 오른쪽 비트 시프트를 수행하면 이진수 1000 0001이 얻어진다. 그런데 이 이진수는 10진수 129를 의미하므로 원래의 3번째 데이터가 129번째의 위치로 이동하게 되는 것을 의미한다.

본 논문에서는 이러한 비트 시프트를 암호화 마스크로 사용하여 데이터의 순서를 셔플링(Shuffling) 하는 비트 시프트 마스크를 소개한다.

3. 제안된 마스크를 이용한 암호화 및 복호화 방식 설명

3.1 데이터의 암호화

본 논문에서는 FFT 버터플라이 알고리즘의 경로 번호를 먼저 선택하는 것을 기본 암호키로 사용하고, 더불어 위에서 제안된 하다마드와 비트 시프트 마스크를 각각 적용 및 복합적으로 사용하여 이미지 데이터를 암호화하는 과정을 설명한다.

위에서 설명한 바와 같이 먼저 FFT의 특정 경로 번호 정보가 기본 암호키로 사용된다. 예를 들어 그림 2에서 보여지는 것처럼 M=16인 경우 4개의 경로가 발생하고 여기서는 경로 번호 1을 기본 암호키로 사용한다.

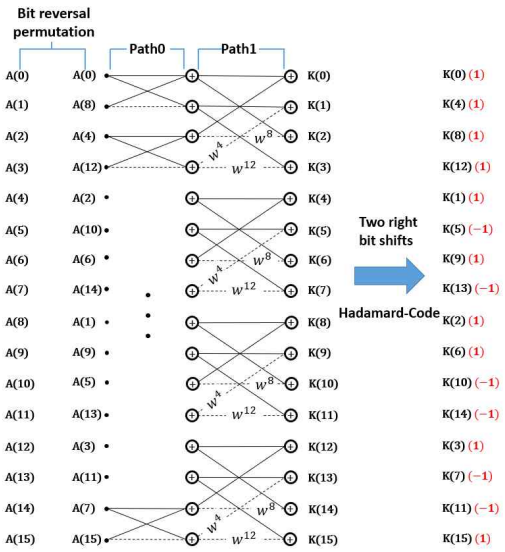

그림 2. M=16일 때 FFT의 버터플라이 알고리즘(일차원). 데이터에 하다마드 코드 및 비트 시프트 적용

Fig. 2. Butterfly algorithm of FFT for M=16 (1D). Application of hadamard code and bit shift to the data

제안된 마스크들을 암호키로 적용하는 방법을 설명하면 먼저 하다마드 마스크는 식 (9)의 4bit 마스크를 사용하였으며 암호키로 사용하기 위해 특정 경로에서 버터플라이 연산이 수행되고 난 뒤의 데이터에 적용이 된다. 그림 2에서 경로 1에서 연산이 완료된 데이터 K(j)에 하마다드 코드를 일차원으로 픽셀 데이터 배열에 복소수 곱하기를 이용하면 하다마드의 원리에 의해 해당 경로의 데이터에 0 또는 π의 위상값을 곱하는 효과와 함께 원본 데이터의 값이 변하게 되며, 이는 픽셀 데이터에 랜덤 페이즈 마스크(Random Phase Mask)를 붙이는 효과를 가져온다. 이러한 하다마드 마스크를 적용하여 암호키로 사용될 수 있다는 것을 보여주고 있다.

다음으로 Bit-Shift를 적용하는 방법을 설명하면, 역시 위의 그림 2를 예를 들어, 경로 1에서 연산이 완료된 데이터 K(j)에 비트 시프트(BS) 방법을 이용하여 데이터 배열의 위치를 바꾼다(Shuffling). 여기서 비트 시프트는 위의 개념에서 설명한 바와 같이 오른쪽, 왼쪽 수행을 하게 되며 본 논문에서는 오른쪽 비트 시프트 수행을 기준으로 수행된 횟수를 암호키로 사용한다. 예를 들어 그림 2 에서는 두 번의 오른쪽 비트 시프트가 수행된 다음의 데이터 배열을 보여준다.

이러한 경우 경로 번호를 기본으로 하는 암호키에 추가로 하다마드, 비트 시프트 암호키를 적용함으로써 기본 2개의 암호키가 한 세트를 구성한다. 이차원(2D) 픽셀 데이터를 암호화하는 경우, FFT는 일차원의 행 데이터와 일차원의 열 데이터에 대해 두 번의 FFT가 수행되는데, 행 데이터와 열 데이터의 암호화에 서로 다른 세트의 암호키가 사용될 수 있다. 이 경우 총 4개의 암호키가 사용된다. 기본 경로 암호키에 하다마드와 비트 시프트를 복합적으로 사용 시 총 6개의 암호키가 사용되게 되며 모든 암호키는 간단한 정수값으로 구성이 된다.

3.2 데이터의 복호화

데이터 복원을 위하여, 암호화된 데이터에 2D 역 고속 푸리에 변환(IFFT, Inverse Fast Fourier Transform)을 수행하며, 이전에 암호화에 사용된 경로 번호, 사용된 하다마드와 비트 시프트 마스크 암호키를 이용해 데이터를 복원한다.

여기서 경로 번호의 암호키로 접근했을 때, FFT의 버터플라이 연산 후 나오는 데이터와 IFFT의 버터플라이 연산 후 나오는 데이터의 위치와 값에 차이가 있으므로 이를 구별하여 복원시켜야 한다.

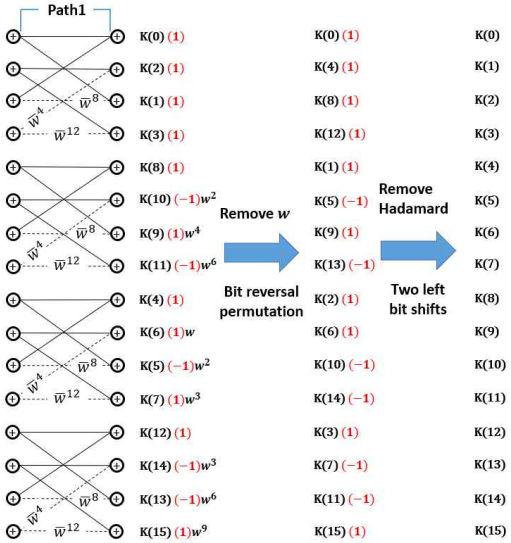

그림 3. 데이터 복원을 위해 경로 1에서 IFFT. 하다마드 코드 적용과 왼쪽 비트 시프트

Fig. 3. IFFT along path 1 for data reconstruction. Application of Hadamard Code and Left Bit Shift

일반적으로 역변환 IFFT에서 각 경로의 데이터 값은 FFT의 해당 경로의 데이터 값과 동일 값으로 연산 될 거라 예상하지만 그렇지 않다.

그림 3에서 보여지는 것처럼, 데이터의 복원을 위해 사용된 암호키 경로 1로 들어오면 연산 된 데이터의 위치 K(j)값이 그림 2와 비교했을 때 다름을 확인할 수 있다. 이를 고려하여 데이터 복원을 위해 먼저 데이터 순서를 맞춰주기 위한 Bit Reversal Permuation(BRP)을 수행하고, 추가적으로 붙어있는 트위들 팩터 w를 제거한다. 그러면 그림 2의 암호화된 데이터와 동일한 배열의 값을 가진 암호화된 데이터를 확인할 수 있고 이 값을 이용하여 본 논문에서 암호키로 사용한 4bit 하다마드 코드를 곱하여 원본 데이터로 복호화한다. 또한 데이터 비트 시프트의 경우, 암호키로 사용한 횟수만큼 여기서는 왼쪽 비트 시프트를 두 번 수행하여 원래의 데이터 순서로 복원시킨다. 그리고 나머지 IFFT 버터플라이 알고리즘을 수행하기 위해 w를 다시 추가하고 BRP를 수행하여 데이터의 배열 순서를 맞춘 후 IFFT를 수행하게 되고 최종 복원된 이미지를 획득할 수 있다.

4. 시뮬레이션

4.1 암호화/복호화 시뮬레이션 데이터

본 논문에서 암호키로 사용한 경로 번호, 하다마드 마스크, 비트 시프트를 이용하여 데이터를 암호화하고 복원해 보았다. 먼저 각 원본 이미지에 대한 일관성 있는 암호 효율을 분석하기 위해 동일한 암호키를 사용하였으며, 각 암호키 세트마다 시뮬레이션을 반복하고 데이터를 이미지화 하였다. 그림 4는 실험에 사용된 원본 데이터를 보여주고 있다. 256×256픽셀 해상도를 가지는 4개의 서로 다른 이미지 Elane, Baboon, Cameraman, Peppers를 사용하여 위의 암호키들을 적용해 보았다. 이차원 이미지로 2D FFT가 수행되며, 수행과정은 먼저 256개의 행을 1D FFT 수행하고 다음으로 256개의 열을 1D FFT 수행한다. 이때 행과 열에 각각 암호키들을 적용하였으며 본 논문에서는 첫 번째 암호키인 경로 번호를 키로 사용한다. 256 픽셀 데이터로 총 8개의 경로가 나오며 이 중 하나를 암호키로 사용하였다. 본 논문에서는 행 데이터에 경로 번호 3과 열 데이터에 경로 번호 5에 암호키를 적용하였다. 두 번째 암호키인 하다마드 마스크는 행 데이터에 4bit 마스크를 열 데이터에 8bit 마스크를 사용하였고, 마지막 키인 비트 시프트는 행 데이터에 3번, 열 데이터에 2번 셔플링을 수행하였다.

암호화된 데이터에 본 연구에서 사용된 정확한 키 정보를 가지고 복원을 하게 되면 그림 5와 같이 복호화된다. 그림 6-8은 각각 하다마드와 비트 시프트, 두 마스크를 복합적으로 적용한 암호화 데이터를 보여주고 있고, 그림 9-11은 잘못된 키 정보로 접근했을 때 복원된 데이터를 보여주고 있다. 다음으로 행 데이터 부분의 암호키가 완전히 노출되었을 때를 가정하여 시뮬레이션을 수행하였고 그림 12-14에서 결과를 보여주고 있다. 위의 시뮬레이션에 사용된 암호키 정보는 표 1-3에 정리하였다.

또한 FFT를 사용함으로써 시뮬레이션 시간은 본 시스템에서 평균 1초 이내로 DFT가 2시간 이상 시뮬레이션을 처리하는데 비해 굉장히 빠른 처리 속도를 확인할 수 있었다. 이는 $M\times N$ 픽셀 데이터를 처리한다고 했을 때, 2D DFT는 원본 이미지를 처리하기 위해 $M^2\times N^2$의 연산이 필요하다. 반면 FFT는 $(MN/2)\log_2 MN$으로 변환 시간이 획기적으로 줄게 되며 256×256 데이터를 예를 든다면 DFT는 2^{32} 연산이, FFT는 8×2^{16} 연산으로 약 8,000배 이상 빠른 처리 속도를 가진다.

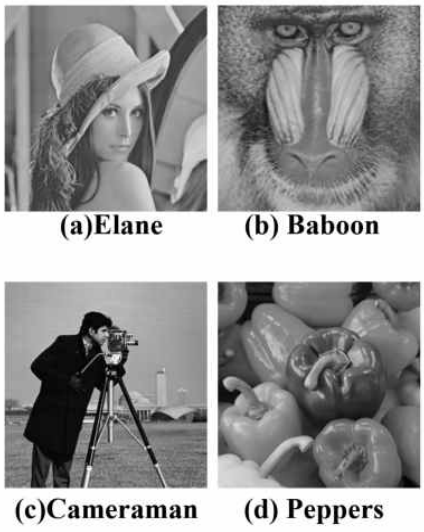

그림 4. 원본 이미지
Fig. 4. Original Images.

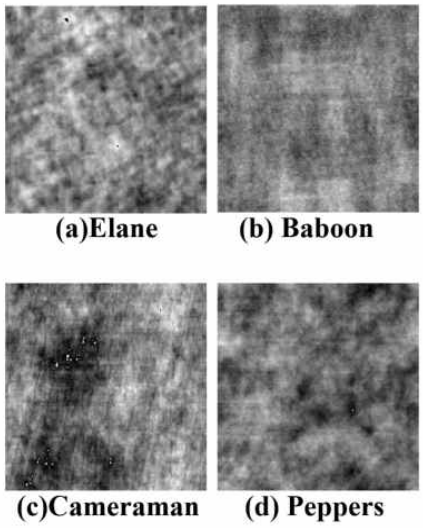

그림 6. 하다마드 마스크를 적용한 암호화 데이터
Fig. 6. Encrypted data with Hadamard mask

그림 5. 복원된 이미지
Fig. 5. Decrypted images

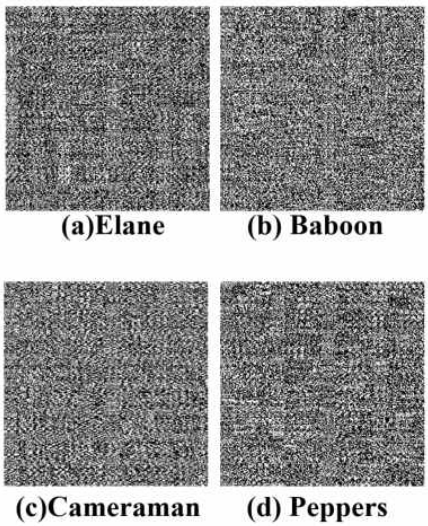

그림 7. 비트 시프트를 적용한 암호화 데이터
Fig. 7. Encrypted data with Bit Shift

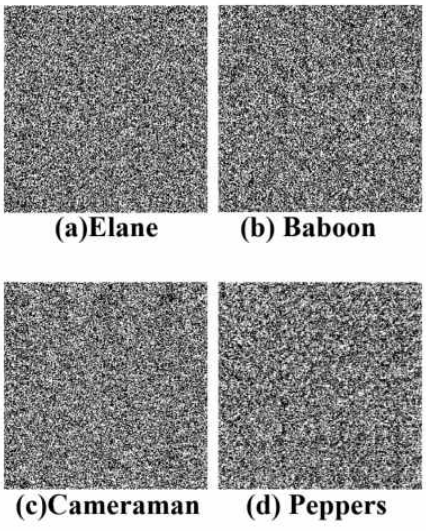

그림 8. 하다마드와 비트 시프트를 동시에 적용한 암호화 데이터
Fig. 8. Encrypted Data with Simultaneous Application of Hadamard and Bit Shift

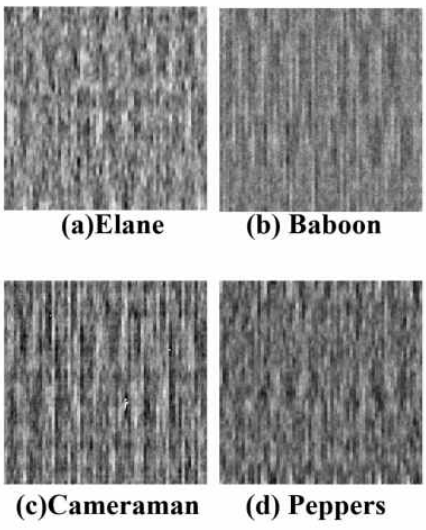

그림 9. 잘못된 하다마드 키로 복원
Fig. 9. Decryption with Incorrect Hadamard key

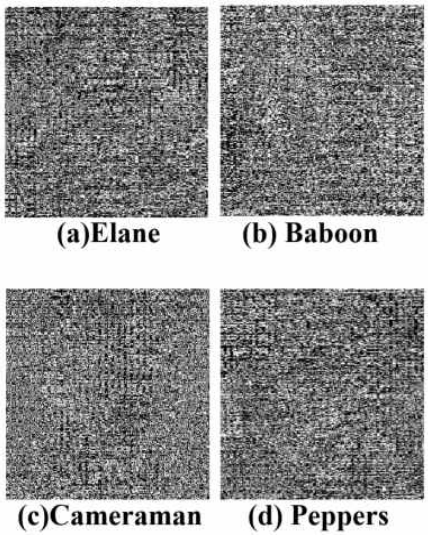

그림 10. 잘못된 비트 시프트 키로 복원
Fig. 10. Decryption with Incorrect Bit Shift key

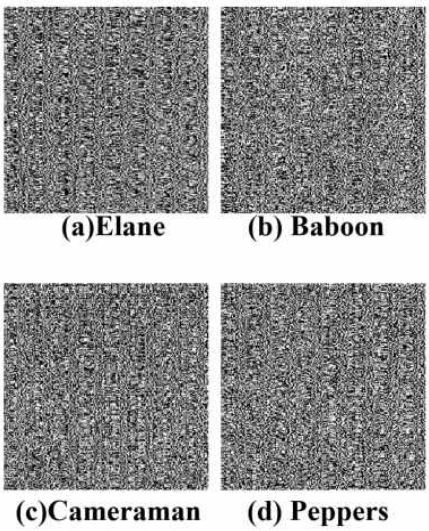

그림 11. 잘못된 하다마드와 비트시프트 키로 복원
Fig. 11. Decryption with incorrect hadamard and bit shift keys

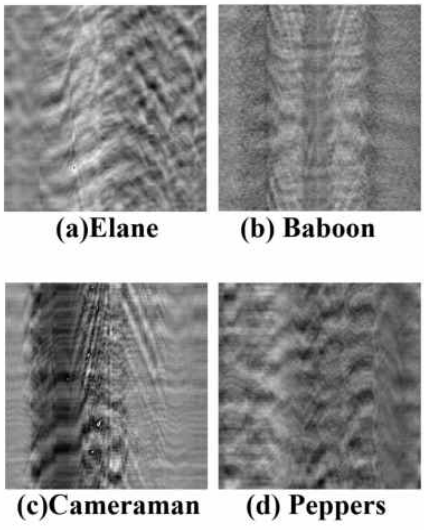

그림 12. 한 부분의 경로, 하다마드 암호키가 완전히 노출 됐을 때 복원

Fig. 12. Decryption with partially correct keys for hadamard

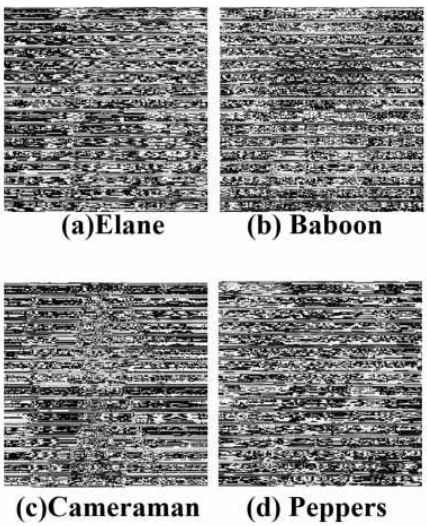

그림 13. 한 부분의 경로, 비트 시프트 암호키가 완전히 노출 됐을 때 복원

Fig. 13. Decryption with partially correct keys for bit shift

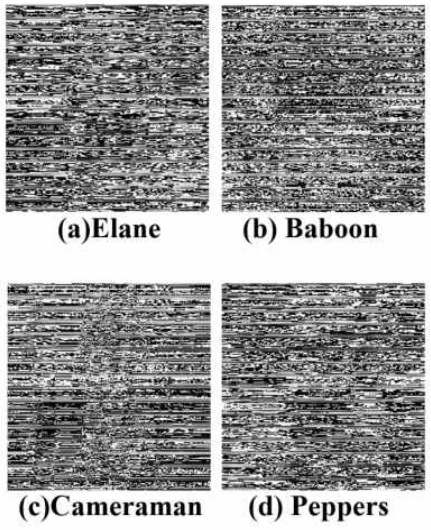

그림 14. 한 부분의 경로, 하다마드, 비트 시프트 암호키가 완전히 노출 됐을 때 복원

Fig. 14. Decryption with partially correct keys for hadamard and bit Shift

4.2 암호화 데이터 성능 분석

위의 실험 결과를 토대로 본 논문에서 제안한 암호화 알고리즘과 암호화 마스크가 데이터 암호화 기술로써 충분한 성능을 가질 수 있는지 분석해 보았다. 본 논문에서는 복호화된 이미지와 원본 이미지 간의 상관관계를 객관적으로 평가하기 위해 상관계수(Correlation Coefficient, *CC*)를 통해 분석하였다. *CC*는 다음 식 (10)과 같이 정의된다.

$$CC = \frac{cov(d, o)}{\sigma(d)\sigma(o)} \qquad (10)$$

여기서 $cov(d, o)$는 복호화된 이미지와 원본 이미지 간의 교차 공분산이고 $\sigma(d)$와 $\sigma(o)$는 각각 복호화된 이미지와 원본 이미지의 표준편차이다. 일반적으로 0.25 미만의 *CC*는 암호화된 이미

지 데이터가 충분히 우수한 암호화 성능을 가지고 있다고 판단하는 허용 가능한 값으로 간주한다[16, 17]. *CC*가 1에 가까울수록 원본 이미지에 가까운 데이터이며 본 연구에서 분석한 결과 복호화된 데이터의 CC는 평균 0.99997로 원본 데이터에 가까운 복원 결과를 확인할 수 있었다. 암호화된 데이터의 *CC*는 표 1에 작성하였으며 결과를 기준으로 각 암호화 마스크를 확인했을 때 하다마드를 이용한 방법은 0.2432, 비트 시프트를 이용한 방법은 0.0143 마스크를 복합적으로 사용했을 때 0.0052의 평균값의 *CC*를 보여주는 것을 확인하였다. 여기서 마스크를 복합적으로 사용했을 때 예상대로 성능이 가장 우수했으며 다음으로 비트 시프트를 이용한 방법, 하다마드 마스크를 이용한 방법 순으로 성능 차이를 확인하였다.

다음으로 잘못된 키 정보를 가지고 복원을 시도했을 때 복원 결과 *CC*를 표 2에 작성하였으며 하다마드는 0.0232, 비트 시프트 0.0161, 둘 다 사용했을 때 0.0041의 평균값이 나오는 것을 확인할 수 있었다. 또한 부분적으로 한 파트(행) 부분의 암호키가 완전히 노출되었을 때 데이터 복원을 시도해본 결과 하다마드는 0.1935, 비트 시프트 0.0488, 둘 다 사용했을 때 0.0031의 평균값의 *CC*가 나오는 것을 확인할 수 있었고 표 3에 작성하였다.

표 1. 이미지 데이터에 사용된 암호화 키 및 CC
Table 1. Encryption keys and CC for image data

FFT / 그림		Elane		Baboon		Cameraman		Peppers	
		행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)
암호화	그림 6. (a)-(d) Hadamard	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5
		4bit	8bit	4bit	8bit	4bit	8bit	4bit	8bit
		CC= 0.2399		CC=0.2435		CC=0.2291		CC=0.2601	
	그림 7. (a)-(d) Bit-Shift	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5
		3 BS	2 BS	3 BS	2 BS	3 BS	2 BS	3 BS	2 BS
		CC=0.0079		CC=0.0259		CC=0.0154		CC=0.0082	
	그림 8. (a)-(d) Hadamard +Bit-Shift	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5	경로 3	경로 5
		4bit	8bit	4bit	8bit	4bit	8bit	4bit	8bit
		3 BS	2 BS	3 BS	2 BS	3 BS	2 BS	3 BS	2 BS
		CC=0.0048		CC=0.0088		CC=0.0047		CC=0.0026	

표 2. 잘못된 암호키 정보로 데이터의 복원과 CC.
Table 2. Decryption with wrong keys

	FFT / 그림	Elane		Baboon		Cameraman		Peppers	
		행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)
복호화 (잘못된 키)	그림 9. (a)-(d) Hadamard	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3
		8bit	4bit	8bit	4bit	8bit	4bit	8bit	4 bit
		CC=0.0268		CC=0.0392		CC=0.0183		CC=0.0086	
	그림 10. (a)-(d) Bit-Shift	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3
		2 BS	5BS	2 BS	5BS	2 BS	5BS	2 BS	5 BS
		CC=0.0176		CC=0.0039		CC=0.0335		CC=0.0097	
	그림 11. (a)-(d) Hadamard +Bit-Shift	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3	경로 2	경로 3
		8bit	4bit	8bit	4bit	8bit	4bit	8bit	4bit
		2 BS	5BS	2 BS	5BS	2 BS	5BS	2 BS	5 BS
		CC=0.0030		CC=0.0017		CC=0.0070		CC=0.0049	

표 3. 암호키가 부분적으로 노출이 됐을 때 사용된 키와 CC; 괄호()는 잘못 사용된 키, 회색 음영은 정확한 키.
Table 3. Partially correct keys. Parentheses denote incorrect key; gray denotes correct key.

	FFT / 그림	Elane		Baboon		Cameraman		Peppers	
		행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)	행(Row)	열(Column)
복호화	그림 12. (a)-(d) Hadamard		(경로 3)		(경로 3)		(경로 3)		(경로 3)
			(4bit)		(4bit)		(4bit)		(4bit)
		CC=0.1251		CC= 0.2570		CC=0.2288		CC=0.1631	
	그림 13. (a)-(d) Bit-Shift		(경로 3)		(경로 3)		(경로 3)		(경로 3)
			(5 BS)		(5 BS)		(5 BS)		(5 BS)
		CC=0.0869		CC= 0.0094		CC=0.0405		CC=0.0586	
	그림 14. (a)-(d) Hadamard +Bit-Shift		(경로 3)		(경로 3)		(경로 3)		(경로 3)
			(4bit)		(4bit)		(4bit)		(4bit)
			(5 BS)		(5 BS)		(5 BS)		(5 BS)
		CC=0.0059		CC= 0.0022		CC=0.0021		CC=0.0021	

5. 결론

본 논문에서는 FFT 버터플라이 알고리즘에서 데이터를 암호화 마스크를 적용하는 기술과 제안된 마스크를 적용하고 결과를 토대로 암호화 데이터로 사용 가능한지 성능 분석을 연구하였다.

버터플라이 알고리즘의 경로 번호를 기본 암호키로 사용하는 것을 특징으로 하며, 이를 기반으로 하다마드와 비트 시프트 마스크를 제작하여 적용하였다. 위의 시뮬레이션을 토대로 성능 분석한 결과, 하마다드 마스크를 이용한 방식은 CC 기준값에 가깝거나 또는 더 높게 나타나는 경우도 있었다. 이는 다른 암호화 마스크인 비트 시프트와 두 마스크를 복합적으로 사용한 실험에 비해 값이 상대적으로 크게 나타나고 있지만 CC 기준으로 봤을 때 준수한 암호화 성능을 보여주고 있다는 것을 확인할 수 있었다.

다음으로 비트 시프트 방식은 우수한 암호화 성능을 보여주고 있다. 이는 경로 중간에 데이터를 셔플링하고 버터플라이 알고리즘을 수행하면서 데이터가 가지는 값이 추측하기 힘은 아주 랜덤한 값을 가지게 된다는 점에서 상당히 우수한 암호화 성능을 보여주고 있다.

암호화 마스크들을 복합적으로 사용한 방식 또한 두 암호화 마스크를 복합적으로 적용함으로써 암호화되는 데이터 값의 복잡성이 올라가게 되고 CC값 또한 데이터 암호화에 아주 우수한 성능을 보여주고 있다고 판단된다.

본 논문에서 사용되는 암호키의 특징은 행 데이터의 암호화와 열 데이터의 암호화가 서로 다른 경로 번호에서 수행될 수 있으며 하나 이상의 여러 경로에서 암호화가 중첩으로 수행될 수도 있다. 또한 경로 번호 암호키의 경우, 데이터의 수가 많아질수록 FFT 경로의 수가 많아져 더 많은 암호키 세트를 구성할 수 있으므로 더 견고한 데이터 암호화를 수행할 수 있다. FFT를 사용하여 빠른 데이터 처리 속도를 보이며 사용된 모든 암호키는 간단한 정수값으로 구성할 수 있는 특징을 가진다.

본 연구를 통해 FFT 버터플라이 연산의 경로에 접근해 다양한 암호화 마스크들을 다중 적용 가능한지 확인하는 실험에서 데이터 암호화 가능성을 확인하였으며, 또한 제안된 암호화 마스크들의 성능 분석을 통해 최적의 암호화 마스크를 선별해보았다. 본 연구를 통해 디지털 데이터를 다루는 다양한 기관에 정보보안 및 암호화 기술에 사용할 수 있다고 기대된다.

참 고 문 헌

[1] P. Refregier, B. Javidi, “Optical image encryption based on input plane and Fourier plane random encoding”, Optics Letter, 20, pp.767-769, Jan. 1, 1995. DOI: https://doi.org/10.1364/OL.20.000767

[2] D. Kong, L. Cao, G. Jin, B. Javidi, “Three-dimensional scene encryption and display based on computer-generated holograms”, Applied Optics, 55(29), pp.8296-8300, Oct. 7, 2016. DOI: https://doi.org/10.1364/AO.55.008296

[3] B. Javidi, A. Carnicer, M. Yamaguchi, T. Nomura, E. Pérez-Cabré, MS. Millán, et al. “Roadmap on optical security”, Journal of Optics, 18(8):083001, Jul. 22, 2016. DOI: https://doi.org/10.1088/2040-8978/18/8/083001

[4] G. Unnikrishnan, J. Joseph, K. Singh, “Optical encryption by double-random phase encoding in the fractional Fourier domain”, Optics Letter, 25(12), pp.887-889, Jun. 15, 2000. DOI: https://doi.org/10.1364/OL.25.000887

[5] Y. Zhang, B. Wang, “Optical image encryption based on interference”, Optics

Letter, 33(21), pp.2443-2445, Oct. 21, 2008. DOI: https://doi.org/10.1364/OL.33.002443

[6] Y. Zhou, K. Panetta, S. Agaian, “Image encryption using discrete parametric cosine transform”, 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, IEEE, pp.395 - 399, Nov, 2009. DOI: http://dx.doi.org/10.1109/ACSSC.2009.5469838

[7] L. H. Zhang, X. Yuan, D. W. Zhang, J. Chen, “Research on Multiple-image Encryption Scheme Based on Fourier Transform and Ghost Imaging Algorithm”, Current Optics and Photonics, 2(4), pp.315-323, Jul. 11, 2018. https://opg.optica.org/copp/abstract.cfm?uri=copp-2-4-315

[8] J. Song, Y. H. Lee, “Optical image encryption using different twiddle factors in the butterfly algorithm of fast Fourier transform”, Optics Communications, 485, 126707, Apr. 15, 2021. DOI: https://doi.org/10.1016/j.optcom.2020.126707

[9] J. Cooley, J. Tukey, “An algorithm for machine computation of complex Fourier series”, Mathematics of Computation. 19, pp.297-301, May. 1, 1965. DOI: https://doi.org/10.2307/2003354

[10] R. N. Bracewell, “The Fourier Transform and Its Applications”, third ed., McGraw Hill, ISBN:9780073039381, 1999.

[11] Y. Zhou, W. Cao, L. Liu, S. Agaian, C.L.P. Chen, “Fast Fourier transform using matrix decomposition”, Information Science, 291, pp.172 - 183, Jan. 10, 2015. DOI: https://doi.org/10.1016/j.ins.2014.08.022

[12] Y. Kim, J. Song, I Moon, Y. H. Lee, “Interference-based multiple-image encryption using binary phase masks”, Optics and Lasers in Engineering, 107, pp.281-287, Apr. 13, 2018. DOI: https://doi.org/10.1016/j.optlaseng.2018.01.012

[13] M. N. Islam, M. S. Alam, “Optical encryption and multiplexing of personal identification information using orthogonal code”, Optical Engineering, 45(9), 098201, Sep. 1, 2006. DOI: https://doi.org/10.1117/1.2354449

[14] JI. Trisnadi, “Hadamard speckle contrast reduction”, Opt Letter, 29(1), 11-13, Jan. 1, 2004. DOI: https://doi.org/10.1364/OL.29.000011

[15] I. H. Lee and M Cho, “Double Random Phase Encryption using Orthogonal Encoding for Multiple-Image Transmission”, Journal of the Optical Society of Korea, 18(3), pp.201-206, Jun. 25, 2014. DOI: https://doi.org/10.3807/JOSK.2014.18.3.201

[16] M. Udovičić, K. Baždarić, L. Bilić-Zulle, M. Petrovečki, “What we need to know when calculating the coefficient of correlation?”, Biochemia Medica, pp.10-15, Jun. 15, 2007. DOI: http://doi.org/10.11613/BM.2007.002.

[17] A.G. Asuero, A. Sayago, A.G. Gonzalez, “The correlation coefficient: An overview”, Critical Reviews in Analytical Chemistry, 36 41-59, Jan. 12, 2007. DOI: https://doi.org/10.1080/10408340500526766

저 자 소 개

이연호(Yeon-Ho Lee)

1980.2 서울대학교 전자공학과 졸업
1989.5 Univ. of Southern California 전기공학-전기물리학 석사, 박사
1991.12 Senior engineer in Aura systems
2022.8 성균관대학교 교수
2022.9-현재 : 성균관대학교 명예 교수
<주관심분야> 정보보안, 이미지 프로세싱, 홀로그램 광암호화

송재훈(Jae-Hun Song)

2011.8 홍익대학교 전자전기공학과 졸업
2014.2 성균관대학교 전자전기컴퓨터공학과 석사
2017.8 성균관대학교 전자전기컴퓨터공학과 박사
2018.7-현재 : 성균관대학교 IT융합연구원 선임연구원
<주관심분야> 정보보안, 이미지 프로세싱, 홀로그램 광암호화