通信情報保護學會論文誌
第6卷 第1號, 1996. 3

고속 암호화 영상처리를 위한 대표성 병렬 시스템 개발+

정 현 수*, 유 은 진**, 전 문 석*, 이 철 희*

The Parallel Encryption System with Representative Theory for High Speed Image Processing

Chung, Hyun-Soo, Yu Eun-Jin, Jun Moon-Seog, Lee Chul-Hee

요 약

본 논문에서는 고속 영상자료를 병렬 암호화할 수 있는 새로운 알고리즘을 제안하였다. 암호화 테이블과 스케닝 순서, 스크램블 등의 문제점들을 개선하였다. 입력자료들은 독립된 블럭으로 분리되며 각 블럭들은 같은 암호화 알고리즘을 통하여 암호화 된다. 그러므로 시스템이 n개의 처리기로 구성되어 있으면, 전체 처리시간이 1/n로 감소됨이 기대된다. 또한 대표성이라는 개념을 적용한 결과, 높은 비도를 갖는 키를 사용한 효과를 얻을 수 있다.

Abstract

In this paper, we suggest a new parallel encryption algorithm for high-speed image processing. We show how to solve problems related to encryption table, a scanning sequence and scraming. The input data is partitioned into independent blocks and the blocks are encrypted by the same cyrptographic algorithms. The total processing time is reduced to 1/n with n processors. In addition, by applying the notion of representativeness, we can get the abstract keys of input data with high-level cryptographic complexity.

1. 개 요

정보화 사회의 진전에 따라 다양한 정보들이 통신망을 통하여 유통되고 있다. 이러한 다양한 정보들에는 크게 음성, 데이터(메일), 영상 정보등이 있으며 대부분이 원문으로 다양

* 숭실 대학교 컴퓨터학부 ** 대유공업전문대학 전산과
+ 본 연구는 95년도 산학협동재단 자원으로 수행된 연구임

한 매체를 통하여 전달되고 있다. 이에 따라 보다 쉽게 정보를 보호하여 전달할 수 있는 방법들에 대한 필요성이 대두되고 있으며 표준화가 진행되고 있다.[1] 소인수분해의 어려움과 이산대수 문제에 어려움을 둔 암호시스템들이 등장하고 있으나 응용 대상이 텍스트 데이터에 국한되고, 정보 보호를 지나치게 강조하여 키 관리에 어려움이 뒤 따르고 있으며 처리시간도 많이 소요된다.[2][3] 이러한 현실로 데이터의 양이 많은 영상 정보에 대한 정보 보호의 욕구를 만족시키지 못하고 있다. 특히, 영상 자료는 일반 데이터와는 달리 상당량의 데이터를 갖고 있어 통신에 많은 소요시간과 저장에 많은 기억공간을 필요로 한다. 따라서 소요시간 단축 및 저장 공간의 감축등의 경제성 문제를 동시에 해결하는 압축 기술이 암호화와 연관되어 많은 연구가 진행되고 있다. 그 결과 대부분의 영상자료 암호시스템들은 영상 압축-암호부호화 방식으로 구현되고 있다.[1][5] 기존의 영상 암호화에서 활용되고 있는 압축 방법으로는 키에 따라 각 심볼마다 비트열을 정의하는 압축테이블을 이용하는 것과 스캐닝 순서 및 스크램블을 이용하여 암호화로 하는 것들이 소개되고 있다.[6][7]

먼저, 압축 방법으로는 통계적 모델을 기본으로 단일 심볼 또는 문자 중심으로 데이터를 압축하는 것으로 이러한 모델에 있어서 압축의 질을 결정하는 요소는 심볼에 대한 정확한 확률을 알아내는 것이다. 통계적 모델에 있어 대표적인 압축방식인 허프만 부호화는 우선 심볼들의 발생확률을 구한 뒤, 이를 바탕으로 빈도수에 반비례하게 심볼에 대한 비트를 부여하여 평균 부호 길이를 최소로 하는 알고리즘이다. 압축 효율은 좋으나 한번 심볼에 대한 비트열이 정의되면 변경이 없고, 심볼에 대한 빈도수를 정의하기 위해서는 먼저 읽어야 하는 것과 압축테이블의 크기로 인하여 보호 관리에 어려움이 따르는 단점이 있다.

사전-기본 압축은 확률에 의존하는 통계적 모델과는 완전히 다른 방법으로 단일 심볼이 아닌 가변 길이의 비트 열을 이용하는 것이다. 토큰이라는 것으로 부호화되면, 토큰은 사전 구문에 대한 색인으로 치환되는 것으로, 구문보다 토큰이 적은 비트로 치환될 때 압축이 이루어 지는 것이다.[8] 사전-기본 압축과 DES를 이용한 압축-암호방식에 있어 먼저, 압축은 입력 자료의 심볼들을 부분열의 문자열 단위로 모아 등장하는 문자열의 테이블에 기억한다. 그 문자열에 해당된 색인으로 문자열을 표현하는 비통계적 압축기법이다. 암호 방식은 압축시 문자열 테이블을 요구하는 점에 착안하여 암호화 키로 문자열 테이블을 사용한다. 그리고 암호문은 문자열 테이블에서의 문자열 부호값, 새로운 문자를 표시하는 부호값, 새로운 문자의 ASCII 부호값들이 DES를 통하여 생성된다. 그러나 DES가 64비트씩 처리되므로 64비트가 될때까지 데이터를 보낼 수 없어 리얼타임 환경에 부적합하다. 또한 압축과 암호과정의 2단계로 수행되어 처리시간이 길어지는 단점이 있다.[8] 또 다른 압축-암호화하는 방법은 압축된 결과를 비트열로 나열하고 초기 비트열은 초기키의 비트열로, 압축열의 비트들을 비밀키로 하여 모듈러 연산을 하거나 초기키 이후의 비밀키 열을 덧셈이나 곱셈 연산으로 모듈러 연산함으로써 암호열을 생성하기도 한다. 단점으로는 평문에 문자를 삽입하거나 문자들의 변경이 어렵고 초기키가 절대적으로 안전하다는 가정하에서만 알고리즘에 대한 신뢰성이 부여된다는 것이다.

따라서 본 논문에서는 압축은 영상 뿐만 아니라 특히 텍스트 자료에 대해서 적용이 가능하게 무손실 압축방법을 적용하였다. 암/복호화는 영상자료를 감안하여 신속한 처리를 할

수 있게 알고리즘을 병렬구조 형태의 시스템인 SIMD(Single Instruction Multiple Data Sets)로 설계, 구현하여 원문정보를 병렬로 암/복호화가 가능하게 하였다. 고정 대표성 구조 암/복호화 알고리즘의 암호화 단계는 먼저, 입력자료들을 고정된 크기씩 읽어들여 분할한다. 그리고 입력 자료로 각각의 구성원들의 속성으로 정의하여 대표성 구조로 만든다. 대표성 구조를 입력으로 압축 효과를 내는 대표성 알고리즘을 통하여 평문을 암호문으로 만든다. 복호화 단계는 암호된 결과들을 다시 원래의 자료로 복원하는 과정으로 암호문을 평문으로 만드는 과정들로 구성되어 있다.

본 논문의 2장에서는 대표성 구조 정의와 평문을 대표성 구조로 만드는 과정들에 대하여 설명하고 있으며, 3장에서는 대표성 구조를 암/복호화하는 알고리즘들과 실제 영상 및 텍스트 자료를 갖고 실험한 결과들을 제시하고 있으며, 마지막으로 결론에서는 고속 병력 암호화 알고리즘의 실험평가와 앞으로의 연구 방향을 제시하였다.

2. 병렬 암/복호화 알고리즘 설계

2.1 대표성 구조 정의

대표성 알고리즘은 구성원들간에 대표성을 갖게 한 대표성 구조에 비트 값들을 구성원 속성들로 정의한 다음, 대표성에 맞지 않는 구성원들을 추출하는 것이다. 이 절에서는 대표성 알고리즘을 이용하여 암호문을 생성하고 복호화하는데 필요한 알고리즘들에 대하여 설명하고 있다.

구성원들간에 대표성을 갖게 하는 구조를 만드는 과정은 먼저 구성원을 몇개의 구성원소로 구성할 것인가를 결정한다. 이에 따라 구성원의 차수가 결정된다. 예를 들어 3개의 구성원소로 구성원을 구성할 경우, 구성원은 (X_2, X_1, X_0)로 표현되고 차수는 3차가 되며 4개로 구성원소를 구성할 경우 구성원은 (X_3, X_2, X_1, X_0)로 표현되고 차수는 4차가 된다.

[정의 1] 구성원이란 $(X_{K-1}, X_{K-2}, .. X_1, X_0)$의 형식을 갖춘 것을 말하며 $X_{K-1}, X_{K-2}, .. X_0$의 각각을 구성원소라 한다.

[정의 2] 차수란 구성원이 몇개의 구성원소로 되었나를 말하는 것으로 $(X_{K-1}, X_{K-2}, .. X_1, X_0)$의 경우 차수는 k가 된다.

차수가 결정되면, 구성원소를 표현하는 값의 범위를 결정하며 이범위에 의해 최대 레벨이 결정된다. 구성원소의 값의 범위를 0부터 $2^m - 1$ $(m > 0)$ 사이로 할 경우, 최대 레벨은 m이 되며 레벨이 m인 구성원은 $(0_{K-1}, 0_{K-2}, .. 0_1, 0_0)$가 된다. 예를 들어 3차이고 구성원소의 값을 0부터 7사이의 값으로 표현했을 때 Σ^3으로 최대 레벨은 3이 되고 최대 레벨 구성원은 (0, 0, 0)가 된다.

[정의 3] 구성원소를 0부터 $2^m - 1$ $(m > 0)$ 사이의 값으로 표현하면 구성원이 될 수 있는 최대 레벨은 m이고, 0이 최소 레벨이 된다. 최대 레벨 구성원은 구성원소 값은 모두 "0"으로 이루어진다.

구성원을 표현한 구성원소의 값에 따라 구성원의 고유 레벨이 결정된다. 3차이고 0에서 7사이의 값으로 구성원의 구성원소를 나타낼 경우에 (0, 0, 0) 구성원 레벨은 (정의 3)에 의해 3이 되고, (4, 6, 7) 구성원 레벨은 0이 된다. 즉, 우측에서 좌측으로 "1"을 만날 때까지의 0의 갯수는 4의 이진수는 "100"으로 두 개이고, 6의 이진수는 "110"으로 한개이며, 7의 이진수

는 "111"로 한 개도 없어 이 중 최소값인 0가 (4, 6, 7)의 구성원 레벨이 되는 것이다. 또한 (4, 4, 2)의 구성원은 4는 2로, 2는 1로 되어 레벨이 1이 된다.[10][11]

[정의 4] 구성원들은 구성원의 구성원소 값에 따라 고유한 레벨을 갖는다. 구성원의 레벨은 구성원의 구성원소들, 즉 X_{K-1}, X_{K-2}, .. , X_0의 값들을 이진수로 표현하여 각각을 우측에서 좌측으로 "1"이 나올 때까지의 "0"의 갯수를 구한다. 이 중 가장 작은 수를 구성원의 레벨로 정의한다.

[규칙 1] 레벨 m - 1 이하의 구성원 (Y_{K1}, Y_{K2}, .. ,Y_0) 생성
레벨이 m인 구성원이 (X_{K1}, X_{K2}, .. , X_0)일 때, 공식 $Z_i = X_i + 2^{m-1}$, $(0 \leq i \geq K-1)$을 이용하여 Z_i를 생성하고 X_i와 Z_i가 2'의 주기로 구성원소
Y_i를 정의하여 구성원 (Y_{K1}, Y_{K2}, .. , Y_0)를 생성한다.

3차이고 최대 레벨 3인 경우 (0, 0, 4)의 구성원은 (정의 4)에 의해 레벨이 2가 되며 이 구성원에 속하는 레벨 1인 구성원의 생성은 규칙 1에 따라 X_i들은 (0, 0, 4)로, Z_i들은 (2, 2, 6)로 정의된다. 이 때 Y_0는 주기사 1로 4, 6, 4, 6, 4, 6, 4, 6이 되며, Y_1는 주기가 2로 0, 0, 2, 2, 0, 0, 2, 2로 되고 마지막으로 Y_2는 주가가 4로 0, 0, 0, 0, 2, 2, 2, 2로 되어 표 1과 같은 구성원들이 만들어진다.

표 1. (0, 0, 4)구성원에 속하는 레벨 1인 구성원

Y_2	Y_2	Y_2	구 성 원	레 벨
0	0	4	(0, 0, 4)	2
0	0	6	(0, 0, 6)	1
0	2	4	(0, 2, 4)	1
0	2	6	(0, 2, 6)	1
2	0	4	(2, 0, 4)	1
2	0	6	(2, 0, 6)	1
2	2	4	(2, 2, 4)	1
2	2	6	(2, 2, 6)	1

표 1에서는 레벨 2인 구성원에 속하는 레벨 1인 구성원을 생성하는 과정을 살펴 보았다. 레벨이 0이 아닌 구성원들은 반드시 레벨 0인 구성원들을 포함하고 있다. 그러므로 레벨이 0이 아닌 구성원에 속하는 레벨 0인 구성원들을 생성하는 것이 필요하다. 예를 들면 표 1에서 레벨 1인 (2, 2, 6) 구성원에 속하는 레벨이 0인 구성원들을 생성하면, 규칙 1에 따라 Xi들은 (2, 2, 6)으로, Z_i들은 (3, 3, 7)로 정의된다. 이 때 Y_0는 6, 7, 6, 7, 6, 7, 6, 7이 되며, Y_1는 2, 2, 3, 3, 2, 2, 3, 3으로 되고 마지막으로 Y_2는 2, 2, 2, 2, 3, 3, 3, 3으로 되어 표 2와 같은 구성원들이 만들어진다.

[정의 5] 레벨이 m(m > 0)인 구성원에 속하는 레벨 m - 1 이하의 구성원들은 (규칙 1)에 따라 생성된다. 레벨이 0이 아닌 구성원에 속하는 레벨 0인 구성원들은 반드시 2^d - 1(단: d = 차수)개씩 생성된다. 그리고 이렇게 묶어 놓은 것을 블럭이라 정의한다. 레벨이 0이 아닌 구성원은 블럭 구성원이라 하며 레벨이 0인 구성원들은 블럭내 구성원이라 정의한다. 블럭 구성원의 속성과 레벨이 블럭 속성과 블럭 레벨로 정의된다.

표 2. (2,2,6)구성원에 속하는 레벨 0인 구성원

Y_2	Y_2	Y_2	구 성 원	레 벨
2	2	6	(2, 2, 6)	1
2	2	7	(2, 2, 7)	0
2	3	6	(2, 3, 6)	0
2	3	7	(2, 3, 7)	0
3	2	6	(3, 2, 6)	0
3	2	7	(3, 2, 7)	0
3	3	6	(3, 3, 6)	0
3	3	7	(3, 3, 7)	0

이렇게 생성된 블럭들은 레벨이 최대인 블럭부터 시작하여 레벨이 m인 블럭에 속하는 레벨 m-1이하의 블럭들은 레벨이 작은 순으로 나열된다. 예를 들면 3차이고 레벨이 3인 경우 먼저 (0, 0, 0)의 블럭이 나열되고 그 다음 (0, 0, 0)에 속하는 블럭중 레벨이 1인 블럭들이 나열된다. 그리고 레벨이 1인 블럭의 나열이 끝나면, 첫번째 레벨이 2인 블럭을 나열한 다음, 레벨 2에 속하는 레벨이 1인 블럭들을 나열한다. 그 다음 레벨이 2인 블럭을 첫번째와 같은 방식으로 나열한다. 이런 순서로 나열한 블럭들은 그림 1로 나타 냈으며 이렇게 나열된 구성원들에 대하여 입력 자료의 비트값을 순차적으로 나열하여 속성을 부여한다. ((그림 1), 참조)

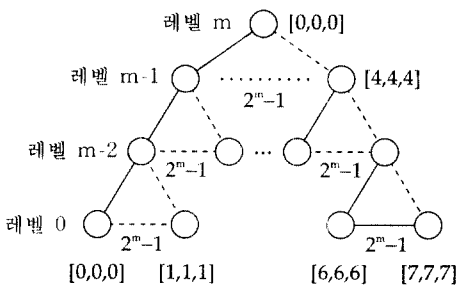

그림 1. 대표성 구조

그림 1에서 실선은 블럭 구성원을 점선은 블럭내 구성원을 나타내고 있다. 실선의 구성원들과 점선들의 구성원들이 2^d개씩 묶어 한 블럭을 구성함을 알 수 있다.

[정의 6] 블럭 레벨 m에 속해 있는 레벨이 m-1 이하의 블럭들은 레벨이 작은 순으로 하여 블럭을 나열하며 맨 좌

측 블럭은 항상 레벨이 최대인 블럭이 된다. 여기서 구성원에 "0" 또는 "1"인 비트 값은 부여한 것을 구성원 속성이라 한다. 이런 순서로 구성원이 나열되고 구성원 속성이 부여된 것을 대표성 구조라 정의한다.

2.2. 대표성 및 구성원 속성 재생 알고리즘

전체 구성원 수를 기준으로 한 블럭내 구성원 수와 구성원을 몇차로 하여 구성원을 정의하는 가에 따라 블럭 구성원 수와 블럭내 구성원 수의 관계를 알아보면 다음과 같다. 전체 구성원 수 2^n을 2^d로 나누면 블럭 구성원 수가 구해진다. 따라서 블럭내 구성원 수는 2^n-블럭 구성원 수가 된다.

[정의 7] 대표성 구조의 특성에 따라 레벨이 m인 블럭 속성이 레벨 m에 속하는 레벨 m-1이하의 모든 구성원 속성을 대표한다. 이를 대표성이라 정의한다. 또한 레벨이 $i(1 \le i \le m-1)$인 블럭 속성을 대표하고 있는 블럭 속성들 중 레벨이 최소인 블럭을 상위블럭이라 한다.

물론 블럭 속성이 블럭내 구성원들의 속성을 대표하고 있다. (정의 5)에 의해 블럭 구성원 레벨은 "0"이 아니며 블럭내 구성원들의 레벨은 "0"이기 때문이다. 대표성 알고리즘이란 대표성 구조에 의해 속성을 대표하고 있는 것을 이용한 것이다. 즉, 대표하고 있는 구성원 속성과 틀린 속성을 갖고 있는 구성원들을 추출해 내는 것이다. 좀더 구체적으로 말하면, 블럭내 구성원들은 상위블럭 속성 또는 블럭 속성과 틀린 구성들만 추출되고, 블럭 구성원들은 상위블럭 구성원과 속성이 틀린 구성원들만 추출되는 것으로 압축의 효과를 이용한 것이다.

【알고리즘 1】 대표성

```
var comp: // 구성원
    b_comp: // 블럭 구성원;
    bi_comp: // 블럭내 구성원
    b_level: // 블럭 레벨;
    m_level: // 최대 레벨
    b_attr: // 블럭 속성;
    c_attr: // 구성원 속성
    u_attr: // 상위 블럭 속성
Begin
    If (comp == b_comp &
                b_level == m_level)
    Output(comp);
    b_attr=c_attr;
    u_attr=c_attr;
Endif;
    If (comp == b_comp &
                u_attr ≠ b_attr)
      Output(comp);
      b_attr=c_attr;
      u_attr=c_attr;
    Endif;
    If (comp == b_comp &
                u_attr == b_attr)
      b_attr=c_attr;
    Endif;
    If (comp == bi_comp &
                b_attr ≠ c_attr)
       Output(comp);
    Endif;
End
```

대표성 알고리즘에 의해 추출되는 구성원들에 대하여 상위블럭 속성과 블럭속성과의 관계에 따라 추출되는 경우만 표 3에 정리하였다. 표 3에 나타나 있듯이 대표성에 반대되는 속성을 갖고 있는 구성원들만 추출되는 것을 알 수 있다.

표 3. 대표성에 따른 구성원 생성

상위 블럭 속성	블럭 속성	추출 구성원
0	0	블럭내 : 속성이 1인 구성원
0	1	블럭구성원, 블럭내 : 속성이 0인 구성원
1	0	블럭구성원, 블럭내 : 속성이 1인 구성원
1	1	블럭내 : 속성이 0인 구성원

대표성에 의한 구성원 생성은 구성원 속성이 현 구성원을 대표하고 있는 블럭 속성 또는 상위블럭 속성과 틀린 경우만 생성된다. 이렇게 생성된 구성원들에 대하여 속성을 "1"로 정의하여 생성된 구성원들을 구별할 수 있는 대표성 구조로 만든다. 이런 대표성 구조를 이용하면 원래의 구성원 속성을 재생할 수 있다. 즉, 구성원 속성이 "1"인 구성원은 상위블럭 또는 블럭 구성원 속성과 반대로, 속성이 "0"인 구성원은 상위 블럭 또는 블럭 구성원 속성과 같게 속성을 정의하면 구성운 속성들이 재생된다.

【알고리즘 2】 구성원 속성 재생

```
var comp: // 구성원
    b_comp: // 블럭 구성원
    bi_comp: // 블럭내 구성원
    b_level: // 블럭 레벨
    m_level: // 최대 레벨
    b_attr: // 블럭 속성
    c_attr: // 구성원 속성
    ac_attr: // 재생 구성원 속성
    u_attr: // 상위 블럭 속성
Begin
    If (comp == b_comp &
            b_level == m_level)
        ac_attr=c_attr;
        b_attr=c_attr;
        u_attr=c_attr;
    Endif;
    If (comp == b_comp &
        u_attr == 0 & b_attr == 0)
        ac_attr=0;
        b_attr=0;
    Endif;
    If (comp == b_comp &
        u_attr == 0 & b_attr == 1)
        ac_attr=1;
        b_attr=1;
    Endif;
    If (comp == b_comp &
        u_attr == 1 & b_attr == 0)
        ac_attr=1;
        b_attr=1;
    Endif;
    If (comp == b_comp &
        u_attr == 1 & b_attr == 1)
        ac_attr=0;
        b_attr=0;
    Endif;
    If (comp == bi_comp &
            b_attr == 0)
```

```
        ac__attr=c__attr;
    Endif;
    If (comp == bi__comp & b__attr == 1)
        c__attr =! c__attr;
        ac__attr=c__attr;
    Endif;
End
```

이에 대한 상세한 알고리즘이 구성원속성 재생 알고리즘이다. 따라서 대표성 알고리즘으로 생성된 구성원들을 식별할 수 있게 속성을 "1"로 정의한 대표성 구조만 있으면 구성원 속성 재생 알고리즘을 통하여 자료들을 재생할 수 있다. 이와 같은 특성을 이용하여 평문을 암호문으로 만들고 다시 암호문을 평문으로 만들 수 있다.

[이론 1] 블럭 구성원이 n개로 구성된 대표성 구조에서 블럭 속성들이 유일하게 정의되면 주기가 2^n을 갖는 비트 스트림들이 대표성 알고리즘에 의해서 생성된다.

(증 명) 이진 트리에서 작성된 비트 스트림이 대표성 알고리즘2에서 생성됨으로 쉽게 증명될 수 있다.

3. 병렬 암/복호화 시스템 구현과 실험

3.1. 고정 블럭 암/복호화 알고리즘

데이터가 많은 영상 자료의 암/복호화에 유용할 수 있는 고정 대표성 구조의 암/복호화 알고리즘에 대하여 설명하고 있다. 고정 대표성 구조란 대표성 구조의 구성원 수를 고정시킨 것을 말한다. 예를 들면, 3차이고 레벨이 3인 경우, 512개의 구성원으로 구성된 대표성 구조가 만들어진다. 즉, 입력 자료들은 512비트씩 읽어 들여 블럭과 블럭내의 구성원 속성으로 정의한다. 이 대표성 구조에서는 블럭 수가 64개로 암호키 값은 최대 2^{64}으로 나타낼 수 있어 10진수 18자리가 가능하다. 또 다른 방법으로는 2^l(단, l=최대 레벨)의 범위 만큼의 2진수를 암호키로 사용하는 것이다. 예를 들면 최대 레벨이 4이면 암호키로 2^{16}을 사용할 수 있다. 사용 방법은 k차일 경우, 구성원소 X_{k-1}값이 n으로 생성된 구성원 속성은 암호키 n번째 값이 "0"이면 그대로, 암호키 n번째 값이 "1"이면 반대로 하여 암호문을 만드는 것이다. 이와 같이 단일 암호화 키를 사용하여 평문을 암호문으로 만들고 암호문을 평문으로 만드는데 활용이 가능한 알고리즘이다.

【알고리즘 3】 고정 대표성 구조 암호화

```
var comp: // 구성원; en__value:
          // 암호키의 n번째 값;
    nb__comp: // n번째 블럭내 구성원;
    c__attr: // 구성원 속성;
    ec__attr: // 암호화된 구성원 속성;
    key: // n비트 암호키
Begin
    Read(x0,x1,・・・xn-1비트);
```

【알고리즘 1】 대표성

```
If (en__value == 0 &
  comp == nb__comp)
  en__attr=c__attr;
Endif;
  If (en__value == 1 &
    comp == nb__comp)
  c__attr =! c__attr; en__attr=c__attr;
Endif;
  Output( key ec__attr );
End
```

고정 대표성 구조 복호화 과정은 고정 대표성 구조 암호화 알고리즘에 의해 생성된 암호문을 입력으로 고정 대표성 복호화 알고리즘에 의해 평문을 만드는 것이다.

【알고리즘 4】 고정 대표성 구조 복호화

```
var comp: // 구성원; en__value:
          // 암호키의 n번째 값
    nb__comp: // n번째 블럭내 구성원
    c__attr: // 구성원 속성; dc__attr:
          // 복호화된 구성원 속성
    ec__attr: // 암호화된 구성원 속성; key:
          // n 비트 암호키
Begin
    Read(x0,x1,・・・xn-1비트);
    Output( key ec__attr );
    If (en__value == 0 &
        comp == nb__comp)
        dc__arrt=c__attr;
    Endif;
    If (en__value == 1 &
        comp == nb__comp)
        c__attr =! c__attr;
        dc__attr=c__attr;
    Endif;
【알고리즘 2】 구성원 속성 재생 수행
End
```

3.2. 팩스통신에 응용

고정 대표성 구조 암/복호화 알고리즘을 팩스통신에 활용할 수 있다. 암/복호화 알고리즘을 하드웨어에 종속 되게 실행하는 방식을 택하면 비도가 높아진다.[9] 즉 팩스의 고유 기계번호를 암호키로 사용하는 것이다. 송신팩스에서는 수신 팩스가 유용한 상태인지를 조회한다. 유용하면, 수신팩스에서는 유용하다는 신호와 수신팩스 고유 기계번호를 단일 비밀키로 암호화하여 송신팩스로 보낸다. 송신 팩스에서는 수신 팩스 고유 기계번호를 단일 비밀키로 복호화하여 암호킬 사용하여 고정 대표성 구조 암호화 알고리즘을 토하여 원문을 암호문으로 만들어 수신팩스에게 보낸다. 수신팩스는 하드웨어적으로 자신의 고유 기계번호를 암호키로 사용하여 수신된 암호문을 고정 대표성 구조 복호화 알고리즘을 토하여 암호문을 평문으로 만들어 출력한다. 이 때 입출력 시간을 줄이기 위해 2^mX 블럭 수 X 2^d씩 한꺼

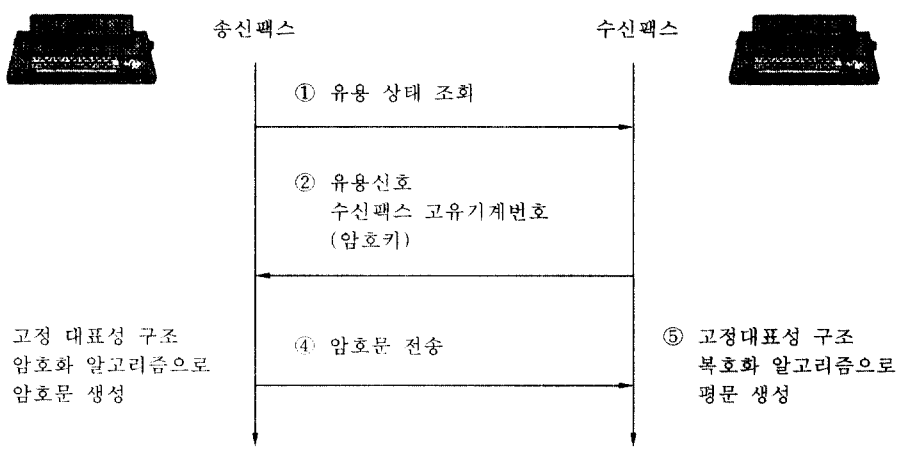

그림 2. 팩스에서의 암/복호화 흐름도

번에 읽어 들여 구성원 식별 대표성 구조를 만드는 것이다. 예를 들면 512비트의 8배씩 한꺼번에 읽어 들인 후, 읽어들인 전체 비트들을 구성원 속성으로 정의할 수 있는 대표성 구조(16x16x16)로 만든다. 그 다음 대표성 알고리즘을 수행한다. 그리고 512비트 단위로 암호키를 이용하여 반복적으로 암호화 과정을 수행하여 암호문을 생성한다. 따라서 병렬로 암호문을 생성하는 경우도 이와 같은 방식으로 처리하면 동일키와 고정 대표성 구조 암호화 알고리즘으로 입력 자료 전체를 암호화할 수 있다.

3.3. 실험 결과 및 분석

가. 영상자료: 256 컬러로 된 영상파일을 입력으로 하여 암호문 및 복호문을 생성한 결과는 (그림 3, 4, 5)와 같다. 이 경우는 512바이트씩 읽어 들여 레벨은 4, 차수는 3인 대표성 구조(16x16x16)로 만들어 암호문을 생성하였다. 결과를 살펴보면 원래 영상과는 아주 다른 영상자료로 암호 영상이 만들어진 것을 볼 수 있다. 또한 암호영상을 복호화한 결과도 원래의 영상과 같게 나와 무손실로 영상이 복원됨을 알 수 있다. 스캔 순서를 변형시키는 것으로는 원 영상자료를 아주 다른 영상 자료로 암호문을 만드는 것이 어려우나 대표성을 이용하면 쉽게 영상 자료를 암호화 할 수 있었다. 또한 암호화된 영상자료를 갖고 원래 영상자료를 추출하는 것도 영상만 분석해 가지고는 불가능하다는 것을 알 수 있다.

그림 3. 원영상 512*512 (256색)

그림 4. 암호화된 영상 512*512 (256색)

그림 5. 복호화된 영상 512*512 (256색)

암호화된 영상을 다시 원래영상으로 복호화 하는데는 그림 7과 8의 내용에서 구현된 실험 결과를 자세히 설명이 연속해서 추가 될 것이다. 실험 결과는 원래 영상의 16진 부호가 암/복화 과정을 통해서 본래의 16진 부호로 구현되기 때문에 그림 5와 같은 동등한 원래영상이 작성됨을 알 수 있다.

나. 공백이 많은 영상파일을 암/복호화한 결과를 16진 부호로 변환한 결과는 그림 6,7,8과 같다. 입력자료의 4,096바이트씩 처리한 결과를 분석해 보면 대표성 구조에 의해] 첫번째 비트 값이 나머지 전체 비트 값들에 영향을 주고 있다. 또한 레벨이 m 이상인 구성원들의 비트 값들이 m-1이하 레벨 구성원들의 비트 값들에 영향을 준다. 따라서 같은 암호키로도 같은 입력 16진 부호에 대해 다른 16진 부호로, 다른 입력 16진 부호에 대해 같은 16진 부호로 암호화된다. 예를 들어 그림 6의 우측 상단에서 3번째 00,00,00,00인 같은 16진 부호들이 암호문인 그림 7에서는

```
00 : 0A 05 01 01 00 00 00 00
     9A 01 28 01 D0 02 5C 01
10 : 00 00 00 FF FF FF CF 18
     CD 6B 95 05 E7 17 01 00
20 : 8A 70 BE 6B 09 08 01 00
     00 7F 00 00 00 00 8A 70
30 : CC 6B 59 08 01 00 7F 05
     00 FF D6 6B 7E 09 00 7F
40 : 00 01 34 00 DD 0F 37 06
     01 00 E6 6B 7E 09 01 00
50 : 00 00 7F 05 90 00 36 06
     FE 6B E0 0B 01 00 00 00
```

```
00 : B4 C4 BE D1 BF DA BF DF
     45 C9 A7 DD 9F D8 13 DE
10 : BF DF BF 20 EA 8A 8F 38
     FC 32 E5 19 D7 E7 71 E0
20 : 35 A0 01 BB B6 D7 BE DF
     8F A3 8F DC 4F DF C5 AF
30 : 73 B4 E6 D7 41 20 3F 25
     70 E3 A6 77 0E E9 70 9F
40 : BF D9 8B DC C8 DF 77 D6
     EC DA 5A B4 39 D1 42 DC
50 : BF DF C0 DA 0F DF B9 D9
     EC 33 A3 17 B9 DA BC DF
```

그림 6. 원 영상의 16진 부호

```
00 : B4 C4 BE D1 BF DA BF DF
     45 C9 A7 DD 9F D8 13 DE
10 : BF DF BF 20 EA 8A 8F 38
     FC 32 E5 19 D7 E7 71 E0
20 : 35 A0 01 BB B6 D7 BE DF
     8F A3 8F DC 4F DF C5 AF
30 : 73 B4 E6 D7 41 20 3F 25
     70 E3 A6 77 0E E9 70 9F
40 : BF D9 8B DC C8 DF 77 D6
     EC DA 5A B4 39 D1 42 DC
50 : BF DF C0 DA 0F DF B9 D9
     EC 33 A3 17 B9 DA BC DF
```

그림 7. 암호화된 영상의 16진 부호

8F, DC, 4F, DF으로 다른 16진 부호로 암호화 되었다. 또한 그림 6의 좌측 가운데 0F, 00인 다른 16진 부호들이 암호문인 그림 7에서는 DF, DF으로 같은 16진 부호로 암호화 되었다.

```
00 : 0A 05 01 01 00 00 00 00
     9A 01 28 01 D0 02 5C 01
10 : 00 00 00 FF FF FF CF 18
     CD 6B 95 05 E7 17 01 00
20 : 8A 70 BE 6B 09 08 01 00
     00 7F 00 00 00 00 8A 70
30 : CC 6B 59 08 01 00 7F 05
     00 FF D6 6B 7E 09 00 7F
40 : 00 01 34 00 DD 0F 37 06
     01 00 E6 6B 7E 09 01 00
50 : 00 00 7F 05 90 00 36 06
     FE 6B E0 0B 01 00 00 00
```

그림 8. 복호화된 영상의 16진 부호

이와 같이 대표성이라는 개념을 적용하여 동일키로 각각 다른 키를 사용한 효과를 나타냈다. 암호문을 입력으로 복호화한 결과는 원래의 자료와 같게 무손실로 복원됨을 알 수 있다. ((그림 8), 참조)

모의실험을 통해서 얻은 결과는 다음과 같다. 65,536개의 암호키를 생성하는데 소요되는 시간을 측정한 결과 2.59초로 한개의 키를 생성하는데 소요되는 시간은 40μs로 나타나 실시간 환경에 적합한으로 보였다. 즉, 대표성 알고리즘은 2^{tn}의 주기를 나타내고 있으면서 키의 생성도 40μs 정도를 나타내고 있어 충분히 실용화 시킬 수 있다고 분석된다. 전체 구성원이 256인 경우 1초에 6,000개, 512인 경우 1,000개 정도의 키를 생성하여 수행된 결과를 얻었다.

4. 결 론

대표성 구조를 이용한 암/복호화 알고리즘은 SIMD 구조를 지원하여 병렬구조에 적합하여 대량의 데이터를 구성된 영상자료를 암호화하고 복호화하는데 매우 유용하다. 물론 텍스트 자료도 처리가 가능하다. 이 알고리즘의 특징으로는 원래의 입력 영역을 고정 크기의 n비트 단위로 분리하여 암/복호화를 처리하므로 n개의 처리기가 있으면 처리 시간을 1/n로 줄일 수가 있다는 것이다. 또한 차수 및 레벨에 따라 다른 대표성 구조가 만들어짐에 따라 차수 및 레벨을 암호키로도 사용할 수도 있다. 이 알고리즘의 가장 큰 특징으로는 하나의 키를 사용하여 암호문을 만들었지만 여러개의 키를 사용한 것과 같은 효과를 나타낸다는 점이다. 이로 인해 키를 찾아내기가 매우 어렵다. 그리고 암호키의 각각이 한 블럭과 블럭내 구성원들의 속성을 정의할 때 영향을 주어 암호키의 전체 비트들의 관계를 정확히 알아야만 복호화가 가능하다. 예를 들어 대표성 구조를 32개의 블럭으로 구성한 경우 2^{32} 가지의 경우의 수가 발생하여 4,294,967,296의 경우를 조사해야 복호화가 가능하다. 블럭의 수가 64개이면 현재의 상대방의 전화번호를 암호키로 사용하여 암호문을 만들 수 있다. 또한 레벨 및 차수를 조정하여 응용 대상에 맞게 암호키의 범위를 결정하여 암/복호화를 수행할 수 있다. 즉 같은 입력에 대해 차수와 레벨이 다르게 대표성 구조를 구성하면 다른 암호문이 생성된다. 실험 결과, 원 영상과는 아주 다르게 영상이 암호화 되기 때문에 영상만 분석해서는 원 영상을 복원할 수 없다. 그리고 암호키 값에 거의 근접한 값으로 복호화를 해도 원 영상과는 아주 다른 영상으로 나타나 암호키를 찾을 확률은 $1/2^n$(단, n: 블럭 구성원수, 구성원소 값의 범위)이다.

앞으로의 연구 방향은 병렬 컴퓨팅 환경에서 시뮬레이션한 결과를 분석해 보는 것이다. 그리고 고정 대표성 구조에 대한 암/복호화 알고리즘을 인증이나 임의의 k 비트 블럭암/

복호화가 가능하도록 개선하고, 이를 인증 메카니즘 및 음성 암호화에 대칭키 방식으로 적용할 예정이다.

참 고 문 헌

[1] Kenji Naemura, Status of Information Security Techniques Standardization in ISO / IEC JTC1 / Sc27, SCIS94-16C.

[2] Introduction to Cryptology, Brian Beckett, Blackwell Scientific Pub., 1988.

[3] 한국전자통신연구소, 현대암호학, 1991

[4] 김용환, "자료압축과 암호화 결합 소프트웨어 구현", 연세대학교 석사 학위 논문, 1992

[5] 정진욱, "압축과 암호코딩의 결합에 관한 연구", 서울대학교 박사학위 논문, 1992

[6] Noboru NAITOH, Image Scramling Scheme Employed a Dyadic Shift, SCIS94-7A

[7] Hayun ROZITAH, A New Scheme on Efficient Scrambling for color Image, SCIS94-7B

[8] The Compression Book, Mark Nelson, M&T Pub., 1992

[9] 박응기(ETRI), "G3 팩시밀리 전송 데이터 보호방법", 한국통신정보보호학회 종합학술발표회 논문집 Vol. 3, No. 1, pp. 215-221, 1993

[10] H. Samet, "Data Structures for quad-tree approximation and compression", Communication ACM 28, 1985, pp. 973-993.

[11] Hyun-Soo Chung, "Optimal OCTREE Algorithms for image Compositions / Decompositions, PRICAI, 1992, pp. 706-712.

□ 著者紹介

정 현 수(Chung, Hyun-Soo)

1982년 2월 숭실대학교 전산학(학사)
1990년 2월 숭실대학교 전산학(석사)
1994년 2월 숭실대학교 전산학(박사)
1982년 3월-현재 ETRI 책임연구원

※ 관심분야 : 영상 및 이동통신, 압축 및 암호화 알고리즘

유 은 진(Yu, Eun-Jin)

1977년 2월 숭실대학교 전산학(학사)
1980년 9월 숭실대학교 전산학(석사)
1993년 3월 숭실대학교 전산학(박사과정)
1984년 3월 한국교육개발원 연구원
1987년 3월-현재 대유공업전문대학 조교수

전 문 석(Jun, Moon-Seog)

1980년 2월 숭실대학교 전자계산학과 학사
1986년 12월 University of Maryland, 전산학 석사
1989년 12월 University of Maryland, 전산학 박사
1989년 1월 ~ 1989년 7월 Morgan State University 전산수학과 조교수
1989년 7월 ~ 1991년 2월 New Mexico State University Physical Science Lab 책임연구원
1993년 3월 ~ 현재 숭실대학교 정보과학대학 부교수

※ 관심분야 : 탐지보안 시스템 설계, Firewall 설계, 병렬 컴퓨터설계, 병렬알고리즘, 암호화 설계

이 철 희(Lee, Chul-Hee)

1958년 2월 육군사관학교 이학사
1962년 2월 Purdue University, 전자공학과 석사
1988년 2월 중앙대학교 전산학과 박사
1962년 9월 ~ 1973년 4월 육군사관학교 전자공학과 조교수
1988년 3월 ~ 1994년 2월 숭실대학교 정보과학대학원 원장
1988년 11월 ~ 1990년 12월 한국정보과학회 회장
현재 숭실대학교 정보과학대학 교수

※ 관심분야 : 데이터 통신, 전산망 구성, 통신망 프로토콜