영상데이터의 안전한 전송을 위한 부분 영상 암호화 시스템 설계

Partial image encryption system design for secure transmission of images

박 시 찬*
Si-Chan Park*

Abstract - This paper proposes partial image encryption system for secure transmission of images. Partial image encryption is suitable for real-time processing purpose of multimedia data that needs compression and encryption. Compression part uses modified SPIHT algorithm and encryption part uses AES. Partial image encryption is significant reduction in encryption time in comparison with whole image encryption.

Key Words :SPIHT, Image encryption, AES

1. 서론

인터넷 환경이 점차 발달하면서 멀티미디어 데이터의 비중이 높아지게 되었다. 그에 따라 유료컨텐츠 및 비밀을 유지해야하는 자료가 증가하면서 그에 대한 보안문제도 중요시되고 있다. 특히 무선 네트워크에서는 그 특성상 유선에 비해 훨씬 가벼운 보안 프로토콜을 적용하고 있어서 상대적으로 쉽게 데이터를 가로채일 수 있으므로 그 문제가 더욱 중요시된다. 그리하여 보안상의 문제를 해결하기 위해 각종 암호 알고리즘을 사용하여 데이터를 보호하고 있다. 그러나 이 과정은 전체 데이터에 대한 암호화 과정으로 동영상과 같은 멀티미디어 데이터에 적용하기에는 시간과 효율성의 측면에서 볼 때 적합하지가 않다. 암호화, 복호화 자체에도 상당한 오버헤드가 있고, 그에 더하여 멀티미디어 데이터의 특성상 방대한 양의 데이터를 전송하며, 고화질의 영상을 구현하기 위해서는 높은 압축률로 압축이 되어야 하기 때문에 압축과정에도 상당한 시간이 소요되어 이러한 두 과정을 실시간으로 처리하기에는 상당한 비용이 든다.

그리하여 본 논문에서는 전체 처리 시간을 줄이기 위해 수정된 SPIHT 알고리즘을 사용하여 부분 영상 암호화 방식을 제안한다. 이 방식은 전체 데이터를 암호화하는 방식보다 처리시간이 상당히 줄어들고 데이터의 보안 측면에서도 떨어지지 않아 영상데이터를 처리하기에 적합하다. 또한 기존의 SPIHT 알고리즘[1]을 수정하여 메모리를 적게 사용하도록 하였고 하드웨어 구성이 가능하게 했다.

본 논문에서는 제 2장에서 SPIHT 알고리즘에 대해서 알아보고 제 3장에서 메모리 사용량을 줄이고 하드웨어 구현이 가능하게 수정된 SPIHT 알고리즘에 대해서 알아보도록 한다. 제 4장에서는 부분 영상 암호화 시스템 설계하였고 제 5장에서 결론을 맺는다.

저자 소개
* 朴 時 贊 : 慶北大學校 電子工學科 碩士課程

2. SPIHT 알고리즘

SPIHT 알고리즘은 EZW(Embedded Zerotree Wavelet) 알고리즘[2]을 개선하여 압축효율을 향상시킨 방법이다. 웨이블릿 변환된 영상의 공간적 자기 유사성을 이용한 zerotree를 이용하여 웨이블릿 계수를 처리하는 점은 EZW와 동일하지만 웨이블릿 계수를 중요도에 따라 부분 집합으로 분할해 가는 과정과 부호화 과정에서 EZW와 차이점을 가진다.

SPIHT 알고리즘은 DWT(Discrete Wavelet Transform) 후에 서브밴드에서의 계수 사이에 트리 구조를 만든다. 그림 1은 웨이블릿 계수 트리를 보여준다.

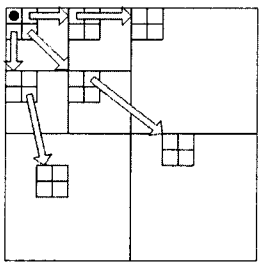

그림 1. 웨이블릿 계수의 트리

SPIHT 알고리즘은 계수 및 트리의 상태를 유지하기 위하여 리스트 구조를 가지고 있으며 중요성 정보는 LIS(List of Insignificant Sets), LIP(List of Insignificant Pixels), LSP

(List of Significant Pixels) 세 개의 리스트에 저장된다. 이 리스트에는 (i, j) 좌표가 저장되며, 이 좌표가 LIP와 LSP 에서는 화소를 의미하지만 LIS에서는 노드의 모든 자손들의 좌표 집합인 D(i, j)를 의미하는 Type-A와 모든 자식들의 좌표 집합을 뺀 L(i, j)를 의미하는 Type-B로 표현된다. 알고리즘의 전체적인 구성은 초기화과정, 분류 과정, Refinement 과정, 양자화 단계 갱신 과정으로 구성되며 세부 알고리즘은 그림 2에 나타나 있다.

1) 초기화
$n = \lfloor \log_2 (max_{(i,j)} \{ |c_{i,j}| \}) \rfloor$
LSP = ∅, LIP = {root node(i, j)},
LIS = { D(root node(i, j)) }
2) 분류 과정
$S_n(\{(i,j)\}) = \begin{cases} 1, & \max\{|c_{i,j}|\} \geq 2^n \\ 0, & otherwise \end{cases}$
For (i, j) ∈LIP
$S_n(i,j)$ 전송
if $S_n(i,j)$==1, (i, j)를 LSP로 이동, $c_{i,j}$의 부호 전송
For (i, j) ∈LIS
if (i, j) == Type-A,
$S_n(D(i,j))$ 전송
if $S_n(D(i,j))$==1,
for (k, l)∈O(i, j)
$S_n(k,l)$ 전송
if $S_n(k,l)$==1, (k, l)을 LSP로 이동, 부호 전송,
else LIP의 끝에 저장
if $L(i,j) \neq \emptyset$
Type-B로서 LIS의 끝에 (i,j)를 옮기고,
Type-B 과정으로 이동
if (i, j) == Type-B,
$S_n(L(i,j))$ 전송
if $S_n(L(i,j))$==1,
Type-A로서 LIS의 끝에 (k, l)을 저장,
LIS로부터 (i, j) 삭제
3) Refinement 과정
For (i, j)∈LSP,
$|c_{i,j}|$의 n번째 MSB 전송
(최근 분류과정(같은 n)에서 포함된 것은 제외)
4) 양자화 단계 갱신
if 0, then end
else n = n -1, 2)번 과정으로 이동

그림 2. SPIHT 알고리즘

3. 수정된 SPIHT 알고리즘

앞에서 살펴본 SPIHT 알고리즘은 상태 정보와 리스트에 필요한 메모리가 많이 요구되어 많은 비용이 필요하다. 또한 자기 유사성을 찾기 위해 동적 자료구조를 사용하기 때문에 외부메모리로의 많은 접근을 요구하여 하드웨어 구현을 어렵게 한다.

그리하여 본 논문에서는 영상의 분할입력방법과 수정된 SPIHT 알고리즘을 사용한다. 먼저 알고리즘의 하드웨어 구현을 가능하게 하기 위해서 원영상을 작은 영상으로 분할하여 입력을 받게 한다. 이것은 전체영상을 하드웨어로 구현할 때는 매우 큰 메모리가 필요하지만 그에 반해 이 방법은 작은 크기의 메모리가 필요하므로 계산에 필요한 메모리를 내부에서 구현하여 처리할만한 크기가 되게 하고, 비용적인 측면에서 상당한 이익을 보게 된다. 또한 데이터의 접근 폭이 줄어들게 되고, 외부메모리로의 접근을 줄이고 내부 메모리 내에서 처리하게 하여 속도 측면에서도 효율적이다. 그리고 원하는 사양에 따라 모듈을 여러 개 병렬로 설계하면 더욱 높은 성능을 얻을 수 있게 된다.

수정된 SPIHT 알고리즘은 원래의 SPIHT 알고리즘에서 상태 정보를 위한 메모리를 줄이기 위해 분류과정과 Refinement 과정을 바꿨다. SPIHT 알고리즘은 Refinement 과정에서 최근에 분류과정에서 포함된 것을 구분하여 제외하기 위해 상태를 저장하는 N^2 bits 메모리(입력된 영상이 N x N이라 가정)가 필요한데 분류과정과 Refinement 과정을 바꾸게 되면 이 정보에 대한 메모리를 절약할 수 있게 된다. 자세한 알고리즘은 그림 3에 나타나 있다.

1) 초기화
$n = \lfloor \log_2 (max_{(i,j)} \{ |c_{i,j}| \}) \rfloor$, {(i, j)| 0≤i,j≤N}
For 0≤i, j≤N-1.
LSP(i, j) = 0;
LIP(i, j) = $\begin{cases} 1, & \text{if } (i,j) \in H \\ 0, & otherwise \end{cases}$
For 0≤i, j≤N/2-1.
LIS(i, j) = $\begin{cases} A, & \text{if } (i,j) \in H \text{and} O(i,j) \neq \emptyset \\ 0, & otherwise \end{cases}$
2) LIP를 위한 Refinement 와 분류 과정
For i=0 to N-1
For j=0 to N-1
if LSP(i, j)==1, $|c_{i,j}|$의 n번째 bit 전송
if LIP(i, j)==1,
$S_n(i,j)$ 전송,
if $S_n(i,j)$==1
LSP(i, j)= 1; LIP(i, j)= 0; $c_{i,j}$의 부호 전송
3) LIS를 위한 분류 과정
For i=0 to N/2 -1
For j=0 to N/2 -1
if LIS(i, j) = A,
$S_n(D(i,j))$ 전송,
if $S_n(D(i,j))$=1
for (k, l)∈O(i, j)
$S_n(k,l)$ 전송
if $S_n(k,l)$=1, LSP(k, l)= 1, $c_{k,l}$부호전송
else LIP(k, l)= 1

```
        if L(i, j)≠∅, LIS(i, j) = B
        else LIS(i, j) = 0;
    if LIS(i, j) = B,
      S_n(L(i,j)) 전송
      if S_n(L(i,j))=1
        for (k, l)∈O(i, j),  LIS(k, l) = A;
        LIS(i, j) = 0;
4) 양자화 단계 갱신
 if n == 0, end
 else n = n-1, 2)번 과정으로 이동
```

그림 3. 수정된 SPIHT 알고리즘

4. 부분 영상 암호화 시스템 설계

본 논문에서는 높은 압축이 되어 있는 영상데이터의 암호화를 위해 기존의 전체 데이터에 대한 암호화 방식 대신 하드웨어 구현이 가능하게 수정된 SPIHT 알고리즘과 AES와 결합한 부분 영상 암호화 시스템을 제안한다. 여기서의 부분 영상 암호화 방법은 영상을 웨이블릿을 통해 변환된 값을 SPIHT 압축 알고리즘을 사용하여 나온 결과에서 상위 두 레벨의 중요성 정보와 초기 threshold 값을 결정하는 n을 암호화하는 방법이다. 이 방법은 부호화 된 데이터의 앞부분에 가장 중요한 정보가 포함되어 있으며 작은 양의 데이터가 달라져도 제대로 복호화 되지 않는 점을 이용한 방법으로 영상 전체를 암호화 한 것과 비교하여 떨어지지 않는다. 그림 4에 시스템의 전체 구조를 나타내었다.

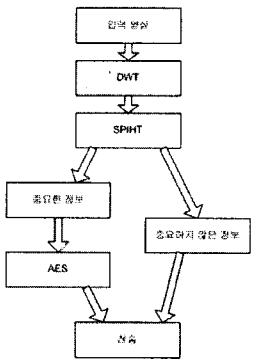

그림 4. 시스템의 전체 구조

본 논문에서는 SPIHT 알고리즘으로 부호화된 정보의 앞부분의 일부를 AES(Advanced Encryption Standard)를 통하여 암호화하였다. 그림 5에 AES의 구조를 나타내었다. AES는 먼저 초기키와 더한 후 ByteSub, ShiftRow, MixColumn, AddRound Key 과정의 라운드를 Nr-1번 반복하고 마지막 라운드에는 MixColmn과정은 제외하고 실행하는 구조를 가진다.

이러한 과정을 통하여 암호화하는 부분영상 암호화 방법은 512 X 512 영상의 경우 암호화해야하는 데이터의 양이 전체의 2%이하였다. 이것은 전체 영상을 암호화하는 것에 비해 상당한 시간 감소의 효과가 있다.

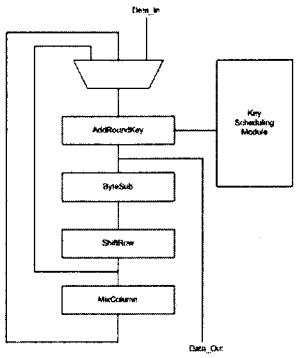

그림 5. AES의 구조

5. 결론

기존의 암호화 시스템은 높은 화질을 위해 높은 압축이 되어 있으며 방대한 양을 가지는 멀티미디어 데이터의 경우에는 영상압축과 암호화의 과정을 실시간으로 처리하기엔 적합하지 않다. 그리하여 본 논문에서는 부분 영상 암호화 시스템을 설계하였다. 압축 알고리즘으로는 중요성을 가진 부분과 중요성이 크지 않은 부분으로 나눌 수 있는 SPIHT 알고리즘을 수정하여 메모리 사용량을 줄이고 하드웨어에도 적합한 알고리즘을 사용하였고 그 결과를 AES암호시스템에 넣어 암호화하였다. 그 결과 이 시스템은 전체 영상을 암호화하는 기존의 시스템보다 상당한 시간 감소가 있었다.

참 고 문 헌

[1] Amir Said, William A. Pearlman, " A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. Circuits Systems for Video Technology, Vol 6, No.3.pp. 243-250, June 1996.

[2] Jerome M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," IEEE Trans. on Signal Proceeding, Vol. 41, No. 12, December 1993

[3] Howard Cheng and Xiaobo Li, "Partial Encryption of Compressed Images and Videos", IEEE Trans. on Signal Processing, Vol 48, No. 8. pp. 2439-2451, August 2000.

[4] Philip P. Dang and Paul M. Chau, "Image encryption for secure internet multimedia application," IEEE Trans. on Consumer Electronics, Vol. 46, No. 3, pp. 395-403, August 2000.

[5] J. Ritter and P. Molitor, "A partitioned wavelet-based approach for image compression using FPGA's.", IEEE Proceedings of the 2000 Custom Integrated Circuits Conference, pp. 547-550, 2000

[6] 서영호, Sujit Dey, 김동욱, "웨이블릿 영역에서의 선택적 부분 영상 암호화", 한국통신학회논문지, Vol. 28, No. 6C, pp.648-657, 2003.6