情報保護學會論文誌
第13券 第3號, 2003. 6

블록 암호를 이용한 안전한 심층 암호

유 정 재*, 김 종 현*, 박 종 혁**, 양 우 일*, 이 상 진*

Secure Steganography Using a Block Cipher

Jeong-Jae Yu*, Jong-Hyun Kim*, Jong-Hyuck Park**, Woo-Il Yang*, Sang-Jin Lee*

요 약

Cachin[1]으로부터 시작된 심층 암호의 안전성에 대한 정의는 Katzenbeisser[2]가 심층 암호의 공격 모델을 기반으로 한 안전성을 논의함으로써 보다 구체화되었으며, 이 후 Hopper[3]가 계산 이론적인 관점에서 일방향 함수가 존재한다면 안전한 심층 암호도 존재함을 증명하였다. 그러나 실제 구현함에 있어 지금까지의 심층 암호 알고리즘은 앞서 정의된 이론적인 안전성을 고려하지 않고, 데이터의 최하위 비트와 비밀 메시지를 치환해 왔다. 비록 데이터의 최하위 비트가 인간의 감각으로는 인지할 수 없으며 랜덤해 보이기까지 하지만 Westfeld[4] 등이 제안한 시각 및 통계 공격에 의하여 비밀 메시지의 삽입을 탐지해낼 수 있었다. 본 논문에서 우리는 블록 암호의 출력이 랜덤한 점을 이용하여 원본을 최소한으로 변형시키지만 충분한 양의 데이터를 삽입하는 방법을 제안하려고 한다. 실험 결과, 제안하는 알고리즘은 단순 최하위 비트 치환 심층 암호와 대등한 데이터 삽입량를 가지면서도 시각 및 통계 공격에 강인하였다.

ABSTRACT

Cachin[1] defined the security of steganography theoretically at first, then Katzenbeisser[2] and Hopper[3] also discussed it on the different aspects. Unfortunately, because many steganographic systems couldn't overcome the statistical gap between a stego-cover and a pure cover, the secure steganography hasn't been evaluated yet. By the effectivel steganalysis algorithm, statistical test which was suggested by Westfeld[4], the attacker Wendy could select the stego-covers out of suspicious covers. Our newly developed algorithm which minimizes the changes of a pure cover by using the block cipher withstands a statistical test and has a similar embedding capacity in comparison with a simple LSB substitution steganography.

Keyword : *steganography, statistical analysis, block cipher*

Ⅰ. 서 론

예전의 심층 암호는 일반 문서에 보이지 않는 잉크로 글을 써 넣는다거나 적당한 길이로 문장을 잘라 재배열하면 실제 전달하려는 문장이 나타나는 등의 물리적인 심층 암호[5]가 대부분이었고, 용도 또한 군사용이거나 범죄에 악용될 여지가 많았으므로 그리 학문적인 연구가 활발하게 이루어지지 못하고 있었다.

그러나 이 후 디지털 통신의 급속한 발전으로 데이터의 복사 및 보급이 용이해짐에 따라 디지털 데이터를 매개체로 한 정보 은닉 분야가 새롭게 관심을 받기 시작했다. 디지털 데이터의 경우 다량의 잉여 부분(redundancy)을 가지고 있으므로 이를 이용하면 대용량의 비밀 정보를 은닉할 수 있게 된 것이다. 따라서 이러한 비밀 통신에 대한 관심도 차차 학문적으로 연구되기 시작하였고, Simmons[6]는 죄수 문제

* 고려대학교 정보보호 대학원(CIST)({shakehds, oracle}@cist.korea.ac.kr, doitnow@hananet.net, sangjin@korea.ac.kr)
** 한화 S&C(주) 기술연구소 연구원(hyuks00@hanwha.co.kr)

를 예로 들어 심층 암호에 대한 논문을 발표하게 된다. 이 논문에서 Simmons는 간수가 죄수들의 통신 내용을 지켜보는 상황하에서도 죄수들이 간수가 눈치채지 못하는 통신로를 구성할 수 있음을 보인다. 이 전까지 보였던 물리적인 심층 암호에 대한 연구가 아니라 디지털 데이터를 매개체로 한 비밀 통신에 대한 가능성을 언급한 최초의 논문인 것이다.

현재 인터넷에서 접할 수 있는 심층 암호들 중 상당수는 디지털 데이터의 최하위 비트(least significant bit)를 비밀 통신용 잉여 부분으로 사용하고 있다. 디지털 데이터 최하위 비트의 단순 치환은 인간의 능력으로 인지하기 어려울 뿐만 아니라 그 용량 또한 충분하며, 구현이 간편하기 때문에 가장 많이 이용되는 방법이다. 또한 이런 식으로 심층 암호를 구성한다면 디지털 데이터의 최하위 비트들이 완전히 랜덤하므로 비밀키를 공유하지 않은 제 3자는 절대로 의미있는 정보가 삽입되었다는 사실을 알 수 없을 것이라는 직관적인 안전성을 믿고 있었다.

그러나 Westfeld[4]등은 실제 디지털 데이터의 최하위 비트들이 완전히 랜덤하게 분포하지 않는다는 사실을 실험하였고 이 결과를 이용하여 효과적인 심층 암호 공격법을 제안하였다. 예를 들어 그림 파일을 매개체로 이용했을 경우, 단순히 색상값(pixel)이나 DCT 변환 계수(coefficient)의 최하위 비트를 변화시켜 통신하는 심층 암호는 시각 공격이나 통계 공격에 의해 탐지해낼 수 있음을 보였다. 단순한 최하위 비트 치환 방식의 심층 암호로는 안전한 비밀 통신을 할 수 없음을 증명한 것이다. 다행히 적은 양의 메시지를 매개체의 전반에 걸쳐 골고루 분산시켜 삽입한다면 위에서 제안한 공격들을 피할 수도 있으나 지극히 제한된 양의 메시지만을 삽입해야 한다는 제약 또한 뒤따르게 되었다. 따라서 단순히 최하위 비트를 치환하는 것만으로는 심층 암호의 안전성과 효율성을 동시에 만족할 수 없었으므로 그 후에 발표된 다른 기법의 심층 암호 도구들도 이 두 가지의 필요 조건 중에서 어느 하나만를 만족시키면서 발전하기 시작하였다.

본 논문에서 우리는 블록 암호의 쇄도 효과(avalanche affect)[1]를 이용한다면 이 두 가지 필요 조건을 동시에 만족시킬 수 있다고 제안할 것이다. 입력의 작은 변화로부터 다양한 랜덤 출력을 얻는 블록 암호의 성질을 이용하면 원본에는 작은 양의 변화를 가하지만 출력값으로부터 많은 양의 메시지를 삽입한 것과 같은 효과를 얻을 수 있는 점에 주목하였다. 실험 결과, 블록 암호를 이용한 심층 암호 알고리즘은 시각 및 통계 공격에 강인하였고 충분한 데이터 삽입도 가능하였다. Hopper[3]가 이론적으로 증명했던 안전한 심층 암호 프로토콜을 우리는 블록 암호를 이용해서 구현한 것이다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 심층 암호의 안전성에 대하여 논의하고 3장에서 단순 최하위 비트 치환 심층 암호가 심층 암호로서 안전하지 않다는 사실을 시각 및 통계 공격을 통해 살펴보겠으며 4장에서는 블록 암호를 이용한 심층 암호의 알고리즘을 설명할 것이다. 5장에서는 구현한 실험 결과를 원본의 통계량과 비교하여 설명하고 6장에서 결론을 맺는다.

Ⅱ. 심층 암호의 안전성

직관적으로 안전한 심층 암호를 정의한다면 수동적 공격자(passive attacker)가 순수한 매개체(원본, cover-data)와 비밀 메시지가 있는 매개체(은닉물, stego-data)를 구별 불가능(indistinguishable)할 때 이러한 심층 암호를 안전한 심층 암호라고 할 것이다.

위와 같은 심층 암호의 안전성에 대한 이론적 정의는 Cachin[1]으로부터 시작되었고, 암호학에서와 마찬가지로 먼저 완전한 심층 암호의 안전성에서 출발하였다. 심층 암호는 비밀키가 필요없는 순수한 심층 암호(pure steganography)와 비밀키를 필요로 하는 비밀키 심층 암호(secret key steganography)로 구분할 수 있는데 비밀키 심층 암호에 대한 정의를 살펴보면 다음과 같다.

정의 2.1 (비밀키 심층 암호)

C가 매개체의 집합, M이 메시지 집합, K가 비밀키 집합일 때, $S=\{C, M, K, E, D\}$를 비밀키 심층 암호계라고 한다. 여기서 E는 메시지 삽입 알고리즘을, D는 메시지 추출 알고리즘을 나타내고 다음을 만족해야 한다.

$$|C| \geq |M|, \quad E: C \times M \times K \rightarrow C, \quad D: C \times K \rightarrow M$$
$$\forall m \in M,\ c \in C,\ k \in K, \quad D(E(c, m, k), k) = m$$

확률 분포 함수 P_C를 가지는 임의의 매개체 C와

1) 평문 또는 키 값의 소량의 변화가 암호문에는 매우 큰 변화를 가져오는 효과

확률 분포 함수 P_S를 가지는 임의의 은닉물, 즉 $S=E(c,m,k)$, 사이의 상대 엔트로피 $D(P_C||P_S)$를 주어진 집합 Q 위에서 식 (1)과 같이 정의한다.

$$D(P_C||P_S)=\sum_{q\in Q}P_C(q)\log_2\frac{P_C(q)}{P_S(q)} \quad (1)$$

여기서 Q는 원본과 은닉물의 확률 분포를 측정할 수 있는 집합을 가르킨다.

정의 2.2 (안전한 심층 암호)

비밀키 심층 암호계 $S=\{C,M,K,E,D\}$에서 P_C가 원본의 확률 분포, P_S가 삽입물의 확률 분포를 나타낸다고 하자. 심층 암호 S가 다음과 같은 성질을 만족하면, 수동적 공격자 W에 대하여 ε 만큼의 안전성을 갖는다고 말한다.

$$D(P_C||P_S)\le\varepsilon$$

특히 $\varepsilon=0$일 때, 심층 암호 S는 완전한 안전성을 가진다 라고 한다.

위의 정의에서 보면 심층 암호의 안전성은 기존의 암호 통신과는 달리 메시지의 내용을 제 3자가 알 수 없도록 숨기는 것(기밀성)보다는 비밀 메시지가 매개체(cover-data)에 삽입되었는가, 아닌가를 구별할 수 없도록 하는 점에 더욱 강조하고 있음을 알 수 있다. 즉 심층 암호의 궁극적인 목표는 비밀 통신의 존재 사실조차 알 수 없도록 하는 것이다. 만약 어떤 심층 암호가 완전한 안전성을 가졌다고 한다면 원본의 확률 분포와 은닉물의 확률 분포가 모두 같아서 통신로 상의 어떠한 데이터를 획득하여도 이 데이터가 원본인지 은닉물인지 구별할 수 없다는 의미이다. 실제로 위에서 정의된 완전한 심층 암호를 구현하자면 Alice는 원본 데이터에 어떠한 변형도 가하지 않고 전달하되, Bob은 이 원본 데이터에 내포된 의미를 파악해야만 비밀 통신이 성립된다. Bob도 전달받은 데이터가 원본인지 은닉물인지 구별할 수 없다면, 또는 구별했다하더라도 은닉물로부터 비밀 메시지를 얻어 내지 못한다면 그것은 더 이상 심층 암호로서의 의미가 없다. 따라서 만약에 단순 최하위 비트 치환 심층 암호를 사용하여 완전한 안전성을 얻고자 한다면 Alice는 무수히 많은 원본 데이터 중에서 Bob에게 전달할 비밀 메시지와 똑같은 최하위 비트를 가진 원본을 골라 전달해야만 할 것이다. 따라서 이러한 심층 암호를 실제로 구현한다는 것은 불가능하다고 볼 수 있다.

이 후에는 보다 현실적인 심층 암호의 안전성에 대한 정의가 이루어지기 시작하는데 Katzenbeisser[2]는 두 가지 오라클을 가정한 심층 암호의 조건적인 안전성(conditional security)을 정의하였다. 암호 통신에서 공격 모델을 가정하고 이러한 공격에 대해 안전한 암호를 설계하듯이 Katzenbeisser도 그런 맥락에서 심층 암호의 안전성을 정의한 것이다.

Katzenbeisser는 먼저 원본 C에 대한 분포를 추정해 주는 오라클 U를 가정한 후, 이 오라클 U로부터 구별 불가능한 원본 데이터 c를 획득할 수 있도록 하였다. 그리고 원본 데이터 c와 비밀 메시지 m, 그리고 키 k를 입력으로 해서 은닉 데이터 s를 출력하는 또 하나의 오라클 V_k를 가정하였다.

$$U(C)=\{c_1,c_2,\ldots\},\ V_k\{E(c,m,k)=s, D(s,k)=m\}$$

이러한 오라클들은 실패 메시지를 반환할 수도 있는 확률적 알고리즘이다. 그리하여 위의 두 가지 오라클을 정해진 q번만 이용하면서, 다항식 시간의 계산 능력을 가지는 공격자에 대한 심층 암호의 안전성을 고려하였다. 만약 어떤 심층 암호로 생성한 임의의 은닉 데이터 s에 대하여 삽입된 메시지가 있는지, 없는지를 공격자가 오라클 U_i과 V_k를 q번 활용하여 정확하게 판단할 확률 $P(k)$가 $\frac{1}{2}$보다 무시할[2]만큼 작다면 그 심층 암호는 오라클 U_i에 대해 안전하다 라고 정의된다. 그리고 모든 오라클 U -가능한 모든 원본 데이터를 출력하는 알고리즘-를 가진 공격자에 대해서 원본과 은닉물을 구별할 수 없다면 그 심층 암호는 조건적으로 안전하다라고 정의하였다. 비록 Katzenbeisser가 엄밀한 정보량이나 계산 복잡도를 고려하지 않고 심층 암호의 안전성을 정의하긴 했지만, 실질적인 심층 암호의 구현에 대하여 안전성을 점검할 수 있는 기준을 제공한 것으로 볼 수 있다.

또한 Hopper[3]등은 심층 암호의 안전성을 계산 복잡도 이론(complexity-theory) 측면에서 정의하였고 일

2) 임의의 k에 대하여 다음을 만족하는 양수 k0와 Nk가 존재할 때 P(k)를 무시할만하다 라고 한다.

$\forall k\ge k_0,\ N_k<\frac{1}{P(k)}$

방향 함수가 존재한다면 안전한 심층 암호도 존재할 수 있음을 증명하였다. 계산 복잡도 측면의 심층 암호 안전성을 정의하기 위해서 몇가지 용어를 정리하도록 하자.

정의 2.3 (통신로)

통신로(channel)란 단조 증가하는 시간 표시가 포함된 비트들의 수열로 다음과 같이 표현된다.

$$\forall i>0,\ t_{i+1}\geq t_i,\ (\{0,1\},t_1),(\{0,1\},t_2),\ldots$$

유한한 공급원 X를 가지는 통신로 C상에서 최소 엔트로피 $H(C)$는 식 (2)과 같이 정의한다.

$$H(C)=\min_{x\in X}\{-\log_2 P_C[x]\} \tag{2}$$

Hopper[6]는 과거의 통신 기록 h를 가지는 통신로 C에서 다음 b개 블록의 분포, C_h^b를 유추할 수 있는 오라클 U의 능력을 식 (3)과 같이 제한하였고

$$\forall h,\ H(C_h^b)>2 \tag{3}$$

이 오라클 U과 일방향 함수 $F:\{0,1\}^k\times\{0,1\}^L\times\{0,1\}^*\rightarrow\{0,1\}^l$을 이용하여 안전한 심층 암호를 구성하였다. 구체적인 알고리즘을 살펴보면, Alice와 Bob 사이에 비밀키와 일방향 함수는 사전에 공유하고 있다고 가정했을 때 비밀 메시지 $m=m_1\|m_2\|\ldots\|m_l$에 대하여

$$U(C_h^b)=c_i,\ F(c_i)=m_j,\ 1\leq i\leq n,\ 1\leq j\leq l \tag{4}$$

를 만족할 때까지 n번 반복 시행한다. 이 과정에서 일치하는 결과를 얻었을 경우 은닉물을 $c=c_1\|c_2\|\ldots\|c_l$로 반환하고, 일치하는 값이 없을 경우에는 실패 메시지를 반환하는 확률적 알고리즘이다.

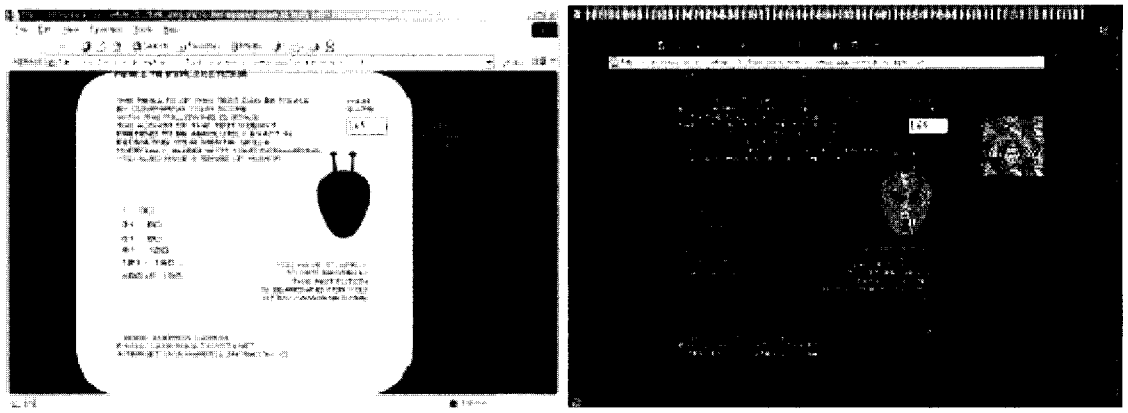
그리고 Hopper[3]는 이 알고리즘의 안전성을 일방향 함수의 안전성으로부터 기인함을 귀류법을 써서 증명하였다. 이로써 적절한 일방향 함수만 존재한다면 안전한 심층 암호의 존재를 증명하였으므로 실질적인 구현도 가능할 것이다. 그러나 아직까지 안전한 심층 암호를 실제로 구현하기에는 무수히 많은 원본과 구별 불가능한 데이터를 얻을 수 있는 오라클과 일방향 함수에 대한 구현이 일단 선행되어야 한다. 우리는 첫번째로 원본에 실험으로 얻은 임계치 이하의 변형만을 가한 데이터를 오라클로부터 획득할 수 있는 데이터라고 가정하였고, 두번째로 일방향 함수의 현실적인 대안으로 기존의 암호 통신에서 안전성이 검증된 블록 암호를 고려해 보았다. Hopper의 알고리즘과 비교할 때 은닉 데이터를 원본과 구별 불가능한 데이터로 전송하는 것 대신에 최소의 변형을 가한 원본 데이터를 은닉 데이터로 전송한다는 차이를 가지게 되겠지만 아직까지 원본 데이터의 전체적인 확률분포에 대해 알려진 바가 없고 이를 추정한다거나 계산하는 일 또한 매우 어려운 작업이므로 실험을 통해 적절한 임계치를 설정한다면 주어진 계산 시간 내에 원본 데이터와 이렇게 얻어진 은닉 데이터를 구별하기란 거의 불가능할 것이다. 그리고 일방향 함수로 안전성이 검증된 블록 암호를 선택할 경우 주어진 계산 시간 내에서는 오직 키에 의해서만 평문을 유추할 수 있으므로 이 키를 심층 암호 통신 송,수신자 간의 비밀로 공유한다면 일방향성 또한 만족하게 될 것이다.

따라서 본 논문에서는 주어진 원본을 작은 블록으로 분할하고 각 블록들을 암호화한 후 삽입하고자 하는 메시지와 일치하면 그 블록에 데이터가 삽입된 것으로 보고 일치하지 않으면 메시지가 삽입되지 않았다고 판단함으로써 Hopper 등이 제안했던 증명 가능한 심층 암호 기법을 실용화시키고자 한다. 그런데 제안하는 알고리즘은 비밀 메시지의 삽입 유무와 삽입할 수 있는 메시지의 양을 증가시키기 위해 각 블록을 약간 변형시켜야 하기 때문에 최악의 경우 시각 및 통계 공격에 의해 탐지가 될 수도 있으므로 실험을 통한 변형의 임계치 설정이 필요하게 된다.

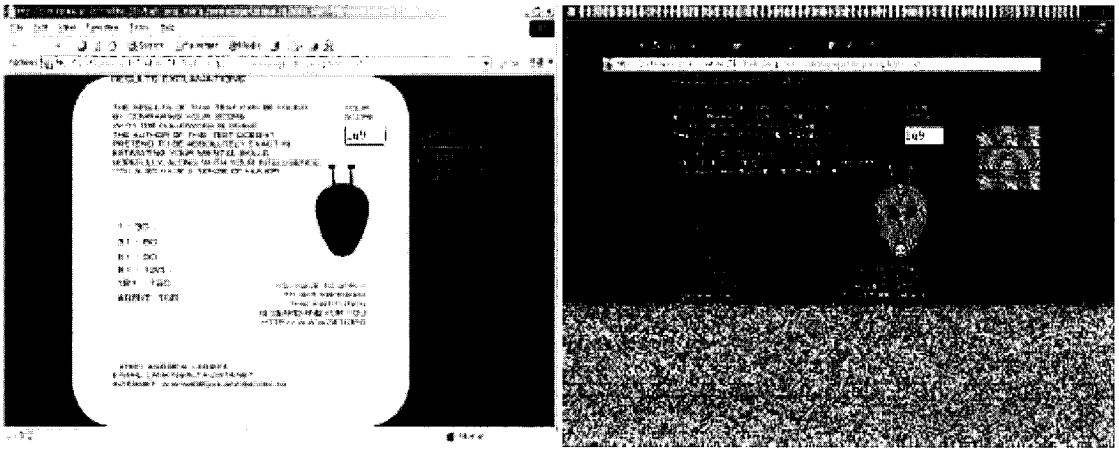
다음 절에서는 가장 널리 알려진 심층 암호의 공격 기법인 시각 및 통계 공격의 원리와 이러한 공격에 대해 심층 암호가 강인하려면 어떠한 조건을 만족해야 하는가에 대해 논의해 볼 것이다.

Ⅲ. 심층 암호의 분석

대부분의 심층 암호 알고리즘들이 가지고 있는 문제점은 디지털 데이터의 최하위 비트가 랜덤한 잉여 부분이라고 생각하고 단순히 원본의 최하위 비트를 비밀 메시지로 치환하여 삽입함으로써 인간의 지각으로는 원본과 은닉물 사이의 차이를 느낄 수 없겠지만 결국 원본이 가지고 있던 통계적 특성을 잃게



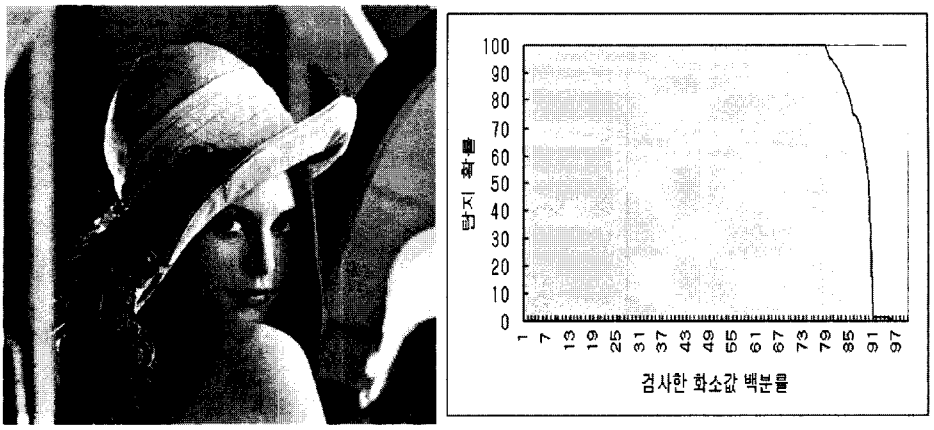
〔그림 1〕 원본(좌)의 최하위 비트 필터링(우)



〔그림 2〕 은닉물(좌)의 최하위 비트 필터링(우)

된다는 것이다. 단순히 외부로 드러나지 않는다고 하여 안전할 것이라고 생각하는 것은 심층 암호로서의 안전성을 포기하겠다는 것과 마찬가지이다. 예를 들어 원본이 그림 파일이고 단순히 최하위 비트를 치환하는 심층 암호의 경우 인간의 시각으로는 원본과 은닉물의 차이점을 발견할 수 없지만 Westfeld[4]등이 제안한 시각 공격이나 통계 공격에 의해 원본과 은닉물을 효과적으로 구별할 수 있었다. 특히 삽입하는 메시지가 암호화되어 있고 삽입량이 많은 경우에는 은닉물의 최하위 비트 분포가 원본의 최하위 비트의 분포를 유지시키지 않고 암호문의 분포처럼 랜덤해지는 성질을 이용한 것이다. Westfeld[2]등은 이와 같은 차이점을 가시화하는 방법으로 시각 공격과 통계 공격 방법을 제시하였다. 시각 공격이라 함은 의심되는 영상을 그 영상 화소(pixel)의 최하위 비트(LSB)만으로 영상을 필터링하는 것을 말하는데 화소의 최하위 비트가 0이면 검정색(RGB=0)으로, 1이면 흰색(RGB=255)으로 치환하는 방식이다. 만약 필터링하는 대상 영상에 아무런 메시지가 삽입되어 있지 않다면 같은 색상의 배경을 구성하는 각 화소들의 최하위 비트들이 일치하여 필터링 후에도 대상의 윤곽이 남아 있지만, 비밀 메시지가 삽입되어 있다면 시각적으로는 같은 색으로 보일지라도 최하위 비트들은 메시지 내용에 따라 변화하여 필터링 후에는 윤곽을 찾아볼 수 없게 된다. [그림 1]은 아무런 메시지도 삽입하지 않은 원본 영상과 그 LSB 필터링을, [그림 2]는 contraband[7]로 전체 LSB의 약 30% 메시지를 삽입한 은닉물과 그 LSB 필터링를 보여준다.

시각 공격이 간단한 조작만으로도 탐지 결과를 육안으로 쉽게 판별할 수 있는 반면, 원본 영상을 손실



〔그림 3〕 단순 LSB 치환 63KB삽입 은닉물(좌)의 통계 테스트 결과(우)

압축 등과 같은 이미지 변환을 하였거나 영상의 윤곽선이 뚜렷하지 못한 경우에는 만족할 만한 결과를 얻을 수 없다. 또한 사람의 육안으로 판별해야 하는 작업인 만큼 객관적인 판단이 어려운 단점도 발생했다. 이에 비해 통계적 공격은 원본과 은닉물 사이에 발생하는 통계적인 차이를 이용한 것이기 때문에 보다 정확한 탐지가 가능하고 탐지 과정을 자동화할 수 있다.

일반적으로 심층 암호는 메시지의 기밀성을 위하여 암호문을 삽입하게 된다. 단순한 최하위 비트 치환 심층 암호라면 이 과정에서 겉으로 드러나진 않지만 원본과 은닉물 사이에 심각한 통계적 차이를 발생시킨다. 메시지를 삽입하지 않은 원본의 인접한 두 색상(PoVs, pairs of values)[3]이 발생한 빈도를 살펴보면 일반적으로 다르게 나타나지만 은닉물에서는 인접한 두 색상의 발생 빈도가 암호문의 통계적 분포 때문에 거의 비슷하게 나타난다. Westfeld[4] 등은 암호문의 랜덤성 테스트[8]에 근거하여 메시지 삽입 확률을 정량화하였다. 이러한 통계적 분석을 정리하면 다음과 같다.

1. 먼저 탐지하려는 그림에서 사용되는 색상들을 팔레트 상의 서로 인접한 두 색상씩 k개의 묶음으로 구분한다.
2. 예상되는 인접한 원본의 색상 빈도, y_i^*를 대상 탐지 파일로부터 구한다.

$$y_i^* = \frac{|n_{2i} + n_{2i+1}|}{2}, \quad 0 \le i \le k-1$$

여기에서 n_{2i}와 n_{2i+1}는 검사하려는 영상의 인접한 두 색상을 나타낸다.

3. 다음과 같이 χ^2값을 계산한다.

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(y_i - y_i^*)^2}{y_i^*}$$

y_i는 n_{2i}나 n_{2i+1}중에서 택일한다.

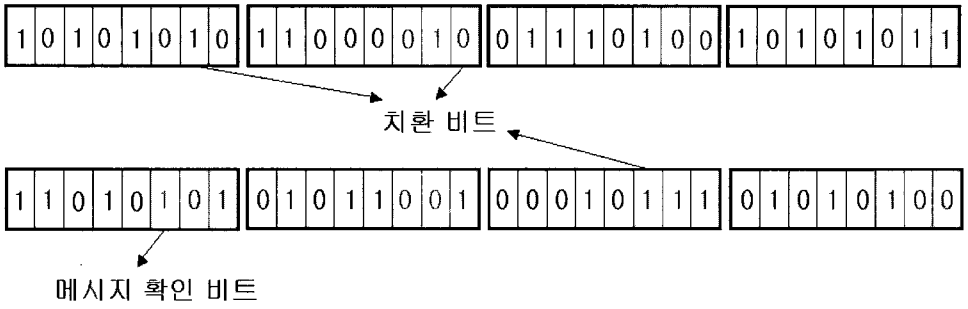
4. y_i와 y_i^*의 분포가 일치할 확률 p는 다음과 같이 가중 분포 함수의 여사건 확률로 주어진다. 여기서 Γ는 Euler Gamma 함수를 나타낸다.

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}} \Gamma(\frac{k-1}{2})} \int_0^{\chi^2_{k-1}} e^{-\frac{x^2}{2}} x^{\frac{k-1}{2}-1} dx$$

[그림 3]은 768KB 원본 영상에 63KB의 메시지를 단순 LSB 치환 심층 암호에 의해 삽입한 은닉물의 통계 테스트 결과를 나타낸다. 그림 우측의 그래프에서 가로축은 탐지 구간의 누적 백분율을, 세로축은 메시지 삽입 확률을 가리킨다. Provos[9]는 탐지 구간을 세분화하거나 PoVs의 순서쌍을 달리하여 χ^2-통계 분석을 확장하였다. 단순 LSB 치환 심층 암호의 경우에는 은닉물의 최하위 비트가 암호문의 통계량을 따르므로 χ^2-통계 테스트로 구별되고 있다.

그러나 위의 통계 분석도 탐지의 오차를 가지고 있었다. 적은 양의 메시지를 원본 그림에 고르게 분

3) 인간의 감각으로는 식별 불가능한 매개체의 인접한 두 데이터 값, 여기서는 그림 파일을 예로 들어 설명할 것이다. 팔레트상의 인접한 두 화소(Pixel)



〔그림 4〕 $l=64$, $r=8$ 로 설정했을 때 분할된 영상 블록

산시켜 삽입하는 경우에는, 오탐지율(false-negative)이 증가한 것이다. 즉 메시지 삽입 후에 판단의 근거가 되는 통계량을 유지시킬 수 있다면 χ^2-통계 분석으로부터 안전할 수 있는 것이다. 이러한 특성을 이용하여 Provos가 PoVs 통계량을 유지하는 심층 암호, OutGuess[9]를 제안하였으나 OutGuess의 메시지 삽입량은 지극히 제한적이었다. 다음 절에서 소개할 블록 암호를 이용한 심층 암호는 χ^2-통계 분석으로 원본의 변화를 감지할 수 없을 정도의 변화만으로 충분한 양의 메시지를 삽입하도록 고안되었다. 원본의 변화를 줄이는 대신 사전 계산을 통해 삽입할 비밀 메시지와 블록 암호의 랜덤한 출력이 일치하는 값을 찾는 과정으로 이루어진다.

Ⅳ. 블록 암호를 이용한 심층 암호

Ⅲ절에서 알 수 있듯이 심층 암호의 안전성과 원본에 대한 변형은 서로 반비례 관계에 있다. 가장 이상적인 경우가 삽입하려는 메시지와 같은 최하위 비트를 가진 원본을 찾는 것이겠지만, 이러한 원본을 찾는 것은 거의 불가능하므로, 현실적인 대안으로 생각할 수 있는 것이 원본의 변형을 최소화해야 한다는 것이다. 이에 대하여 Chandramouli[10] 등은 안전한 심층 암호 통신을 위한 원본의 최하위 비트 변형 비율에 대해 실험적 결과를 바탕으로 논의하였는데, 전체 LSB의 약 33%이하만을 랜덤하게 삽입한다면 χ^2 통계적 분석으로 은닉물을 구별해낼 수 있을 확률이 50%이하로 떨어진다고 하였다. 즉, 임계치 이하의 메시지를 고르게 분포하여 삽입한다면 통계적 분석으로 얻은 결과가 은닉물과 원본의 판단을 랜덤하게 결정하는 확률보다 나을 것이 없다는 것이다. 그래서 우리는 이러한 제한을 만족하면서도 삽입할 수 있는 메시지의 양을 증가시키기 위하여 입력이 한 비트 변하면 출력이 랜덤하게 변하는 블록 암호의 특성을 심층 암호에 도입하였다. 5장의 실험결과에서 다시 언급하겠지만 제안하는 심층 암호는 한 비트 플레인당 최대 25%이하의 변형만을 주면서도 단순 LSB 치환 심층 암호와 대등한 삽입 용량을 가지고 있다. 또한 제안하는 심층 암호는 원본 데이터에 대해 특정 형식-그림, 동영상, 소리 등-에 구애받지 않지만 실험의 편의상 그림 파일을 대상으로 구현하였다. 알고리즘을 살펴보면 다음과 같다.

1. 먼저 원본 데이터 $C=c_{11}\ldots c_{1l}\|c_{21}\ldots c_{2l}\|\ldots$를 적절한 l비트 단위로 분할하여 묶는다. 얼마만한 크기의 비트들로 분할할 것인가는 상황에 따라 적절히 조절하면 되는데 보통 블록 암호의 입력 크기인 64비트에서 128비트 사이의 크기로 분할한다. 이렇게 분할된 블록을 원본의 큰 데이터 블록이라고 하자. 그리고, 각 l비트 블록 내에는 적어도 2개 이상의 치환 가능 비트-큰 데이터 블록 내에 있는 작은 데이터 블록(실험시에는 화소를 사용했음)의 하위 비트들 중에서 선택-가 존재하도록 세부 분할하여, 이 중 한 비트는 반드시 메시지 확인 비트로 사용하고 나머지 비트는 메시지 삽입에 사용하도록 한다.
2. 삽입하려는 메시지 $M=m_1\|m_2\|\ldots\|m_z$도 적당한 크기의 r비트 블록으로 분할한다. 4비트에서 8비트 정도가 적당할 것이다.
3. 비밀키를 입력 값으로 해서 의사난수열(pseudorandom number)을 발생하고, 메시지 확인 비트의 위치를 결정한다. 예를 들어 $l=64$, $r=8$이고 치환 가능 비트가 RGB 화소값의 최하위 세 비트일 때 각 블록은 8개의 화소값을 가지는데 의사난수값

블록 암호를 이용한 메시지 삽입 4.1	블록 암호를 이용한 메시지 추출 4.2

Input : $K=\{0,1\}^k$, $C=c_{11}\ldots c_{1l}\|\|c_{21}\ldots c_{2l}\|\|\ldots$,
$M=m_1\|\|m_2\|\|\ldots\|\|m_z$, $0\le\|m_j\|\le 2^r$
for $j=1,2,\ldots$
$t=rand(K) \bmod l$
$s'_{jt}=c_{jt}=1$; $tag=0$
temp = $c_{j1}\ldots c_{ji}\ldots c_{jl}$
for $i\leftarrow 1$ **to** l **do**
if $E_k(\text{temp}) \bmod 2^r=m_j$
$s_{j1}\ldots s_{ji}\ldots s_{jl}$ = temp
$tag=1$ **break**
else
temp = $c_{j1}\ldots c_{ji}\ldots c_{jl}$
temp = $c_{j1}\ldots c_{j(i-1)}c'_{ji}\ldots c_{jl}$
if $tag=0$
$s'_{jt}=0$
Output : $S=s_{11}\ldots s_{1l}\|\|s_{21}\ldots s_{2l}\|\|\ldots$
	Input : $K=\{0,1\}^k$,
	$S=s_{11}\ldots s_{1l}\|\|s_{21}\ldots s_{2l}\|\|\ldots$
	for $j=1,2,\ldots$
	$t=rand(K) \bmod l$
	if $s'_{jt}=1$
	$E_k(s_{j1}\ldots s_{ji}\ldots s_{jl}) \bmod 2^r=m_j$
	Output : $M=m_1\|\|m_2\|\|\ldots\|\|m_z$

c'_{ji}는 c_{ji} 의 치환 가능 비트를 0 또는 1로 치환했음($c'_{ji}=c_{ji}\oplus 1$)을, s'_{jt}는 메시지 확인 비트를 나타낸다.

이 12라면, 사용되는 치환 가능 비트(0~23) 중 13번째가 메시지 확인 비트가 된다. ([그림 4] 참조).

4-1. 과정 3에서 결정된 원본의 큰 데이터 블록 내 메시지 확인 비트를 1로 치환한 후 이 값을 블록 암호 E로 암호화하여 출력값 $E_k(c_{11}\ldots c_{1i}\ldots c_{1l})$을 얻는다. 그리고, 이 값의 최하위 r 개 비트들만을 취하여 삽입하려는 메시지의 첫번째 블록 m_1과 비교한다. 이 두 값이 일치하면, 다음의 큰 데이터 블록 $(c_{21}\ldots c_{2i}\ldots c_{2l})$으로 이동하고 알고리즘을 처음부터 반복한다.

4-2. 만약 위의 과정 4-1 에서 생성된 출력값의 최하위 r 비트들이 삽입 메시지 m_j와 일치하지 않는다면, 큰 원본 데이터 블록 내 첫번째 치환 가능 비트 중 하나를 변화시키고 과정 4-1를 반복하는데 이 때에도 두 값들이 일치하지 않는다면 치환했던 비트를 원상태로 복구하고 나머지 치환 가능 비트들를 차례대로 변화시키면서 과정 4-1를 다시 반복한다. 큰 원본 데이터 블록 내 마지막 치환 가능 비트를 변화할 때까지도 일치하는 값이 발생하지 않는다면 메시지 확인 비트를 0으로 치환하고 다음의 큰 원본 데이터 블록 $(c_{(i+1)1}\ldots c_{(i+1)i}\ldots c_{(i+1)l})$으로 이동하여 알고리즘을 처음부터 반복한다.

5. 마지막의 큰 원본 데이터 블록 이전에 메시지 삽입이 완료되면, 남은 큰 원본 데이터 블록의 메시지 확인 비트는 모두 0 으로 치환하고 마지막의 큰 원본 데이터 블록 후에도 삽입할 메시지가 남았으면 메시지 삽입이 완전히 이루어지지 않았음을 표시한다.

6. 수신자는 공유된 비밀키로 의사난수열을 발생시켜 메시지 확인 비트의 위치를 결정하고, 삽입물의 데이터 블록을 따라 암호화하면서 메시지를 추출한다. 메시지 추출 시 확인 비트가 0이면 다음의 큰 데이터 블록으로 이동하고, 1이면 이 큰 데이터 블록을 입력으로 블록 암호화하여 출력을 얻은 후 하위 r 비트 메시지 m_j를 추출한다.

$$E_k(s_{j1}\ldots s_{ji}\ldots s_{jl}) \ (\bmod 2^r)=m_j$$

이 과정을 전체 영상에 걸쳐 반복 실행하면 $M=m_1\|\|m_2\|\|\ldots\|\|m_z$을 얻는다.

이 심층 암호의 원리는 간단하다. 원본 데이터에 되도록 적은 변형을 고르게 주면서 충분한 정도의 메시지 삽입량을 얻기 위해서 사전 계산량을 늘린

것이다. 이로 인해 심층 암호가 가지는 성질은 매우 향상되었다.

기존의 LSB 치환은 8 개의 화소(64비트)에 8 비트를 삽입할 수 있으며 최대 8 비트, 평균 4 비트를 변경시켜야 한다. 반면 제안하는 방식은 사용자가 요구하는 안전성과 실용성에 따라 변경되어질 비트 수를 다르게 선택할 수 있다. 효율성을 비교할 수 있도록 LSB 치환과 똑같은 크기의 원본 입력 블록과 삽입 메시지에 대하여 다음 절의 실험 결과를 예로 들어 보자.

64비트 입력 크기를 갖는 블록 암호를 사용하고 각 화소 당 최하위 3비트를 치환 가능 비트로 고려한다. 1 비트의 메시지 확인 비트와 3 비트의 치환 비트를 사용한다면 이 때 암호화된 출력과 메시지의 충돌 확률이 약 92%이므로 최대 4 비트의 변경만으로 8 비트의 메시지를 삽입할 수 있게 된다. 따라서 같은 양의 메시지를 삽입하는 경우 원본이 최대로 많이 변경된다할지라도 변경되는 비트의 개수를 기존의 LSB 치환에 비해 절반 이하로 줄일 수 있는 것이다.

원본의 변화량이 적어질수록 심층 암호의 공격에는 안전하게 될 것이므로 삽입 메시지의 양이 작으면서, 보다 높은 안전성을 얻고자 하는 경우에는 치환 가능 비트의 개수를 줄이면 된다. 더욱이 랜덤하게 발생된 난수열에 의해 메시지 확인 비트가 결정되고 변경되는 비트의 위치 역시 랜덤하므로, 메시지의 특성과는 무관하게 변경된다. 기존의 LSB 치환 방식이 삽입되는 메시지가 암호문일 경우 암호문이 갖고 있는 난수성 때문에 탐지되었던 것에 비해 본 방식은 메시지 특성과 무관하게 원본이 변경되는 특성이 있어 비밀키를 알지 못하면 삽입하는 메시지의 특성(암호문 혹은 평문) 조차도 알 수 없다.

제안한 방식의 심층 암호를 공격하는 가장 단순한 방법은 블록 암호의 키를 전수 조사하는 것이다. 하지만 삽입된 메시지가 암호문일 경우 추출된 데이터가 올바른 값인지 판단하기 위해서는 다시 암호를 해독하는 과정이 필요하기 때문에 이중 암호화와 같은 효과도 가진다. 메시지의 삽입 여부를 판단할 수 있는 다른 방법으로는 주어진 은닉물이 원본이 갖고 있는 일반적인 특성에서 벗어나 있는지 조사하는 방법도 고려해 볼 수 있겠으나 원본의 일반적인 통계 특성은 알려져 있지 않으므로 공격하는 것이 매우 어려운 작업이 된다.

삽입 메시지의 양을 증가시키기 위해서는 치환 가능 비트 수를 증가시키면 되고, 이 경우에도 변경되는 비트 수가 적도록 헤밍 가중치가 작은 것부터 적용시킨다.

V. 실험 결과

블록 암호로 DES를 사용하였고, $n=64, m=8$, 8 비트 화소값의 최하위 1~3 비트중 하나를 메시지 확인 비트로, 나머지 23 개의 치환 가능 비트 중 세 개를 메시지 삽입에 이용한다. 변화되는 비트가 하나의 비트 플레인(bit-plane)에 집중되지 않도록 각각 서로 다른 하위 비트 플레인에 분산시켜 삽입한다면, 한 비트 플레인당 많아야 2개의 치환 비트를 가지게 된다. 예를 들어 4개의 치환 비트(메시지 확인 비트 1+치환 가능 비트3)를 설정할 경우 3개의 비트 플레인에 할당할 치환되는 비트의 경우의 수는 4개의 공을 2개, 1개, 1개의 묶음으로 나누는 경우의 수와 같아진다.

이것은 한 비트 플레인에서 집중적으로 변화가 발생할 확률이 2/8=25%를 넘지 않음을 말해준다. 이미 언급한 바와 같이 Chandramouli[10]의 결과에 의하면 33%이하의 랜덤한 분포는 통계적 분석으로 탐지되지 않는다.

따라서 메시지 확인 비트를 고정하고 난 후, 입력 블록에 변화를 주지 않고 그대로 충돌쌍을 찾을 경우, 한 개의 치환 가능 비트를 변화시킬 경우, 두 개의 비트를 변화시킬 경우, 세 개의 비트를 변화시킬 경우로 분류해서 블록 암호 입력값의 변화를 살펴보면, 각 입력 블록당 최대 648번의 서로 다른 입력을 발생할 수 있다.

$$1+\binom{23}{1}+\left[2\binom{7}{1}\binom{8}{1}+\binom{8}{1}\binom{8}{1}\right]$$

$$+\binom{7}{1}\binom{8}{1}\binom{8}{1}=1+23+176+448=648$$

주어진 입력값에 대하여 DES 연산 후 발생하는 출력값이 랜덤하다고 가정하면 한 번의 DES연산에서 충돌쌍을 찾을 확률은 $(\frac{1}{2})^8$이다. 따라서 648번의 연산 내에 충돌쌍을 찾을 확률은 약 $1-\left(1-(\frac{1}{2})^8\right)^{648}$ ≈ 0.920834이므로 만약 전체 영상의 모든 화소값 최하위 비트를 치환하는 심층 암호를 100%로 본다면 계산상 LSB 치환 대비 약 92%정도를 메시지 삽입에 활용할 수 있게 되는 것이다. 일반적으로 메시지의

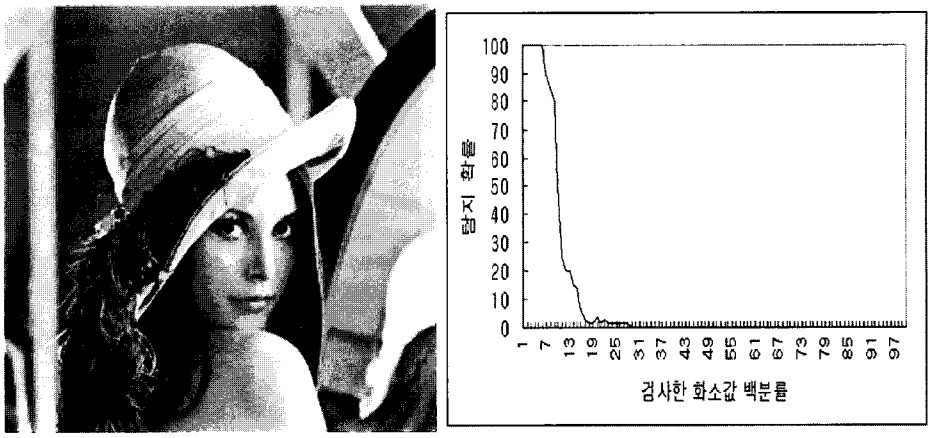
〔그림 5〕 936KB 원본(좌)과 최하위 비트 필터링(우)

〔그림 6〕 63KB 메시지를 삽입한 은닉물(좌)과 최하위 비트 필터링(우)

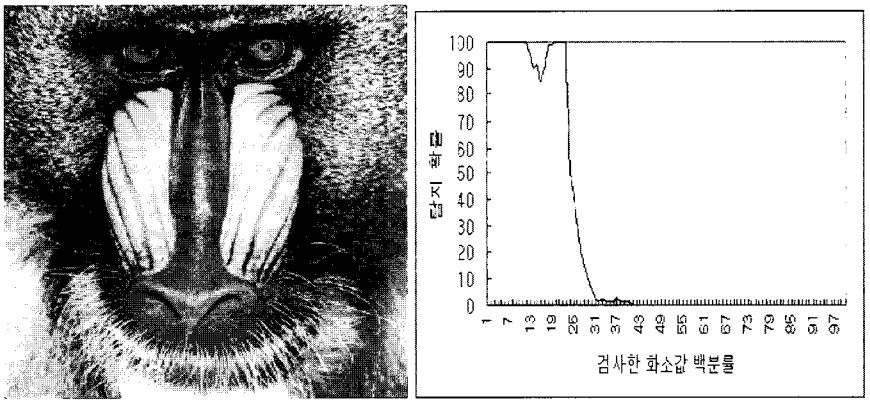
최대 삽입량은 사용되는 원본 데이터와 비밀키에 의존하게 되므로 다소 차이가 있을 수 있겠지만 일단 삽입에 성공하면 χ^2-통계 분석에 의해 탐지되지 않았다.

먼저 [그림 5]는 원본 영상의 시각적 최하위 비트 필터링이고 [그림 6]는 블록 암호를 이용한 심층 암호로 63KB의 메시지를 삽입한 영상의 시각적 필터링이다. 63KB 메시지 삽입 후에도 원본의 분포를 유지하고 있음을 시각적으로 알 수 있다.

영상에 따라 원본이라 할지라도 통계 테스트 결과, 메시지 삽입 가능성이 높은 것으로 나타날 수도 있으나 이것은 원본 영상의 특성을 반영한 것일 따름이다. 검사할 영상이 손실 압축 후 복원된 영상이라면 압축 시 손실되었던 최하위 비트가 랜덤하게 복원되므로 삽입 확률이 높게 나타날 수도 있다. 따라서 주의 깊게 봐야할 점은 바로 블록 암호를 이용



〔그림 7〕 63KB 메시지를 삽입한 은닉물(좌)의 통계 테스트 결과(우)



〔그림 8〕 63KB 메시지를 삽입한 은닉물(좌)의 통계 테스트 결과(우)

한 심층 암호로 메시지를 삽입했을 경우 원본의 통계적 특성과 은닉물의 통계적 특성이 서로 유사하여 어느 것이 원본인지 은닉물인지 확연히 구별할 수 없다는 사실이다. [그림 7, 8] 참조

〔표 5.1〕 원본과 은닉물의 신호 대 잡음비

	신호 대 잡음 비(dB)
Baboon.bmp(그림 8)	46.751
Lena.bmp(그림 7)	46.625
JJH1.bmp(그림 6)	47.108
JJH2.bmp(그림 9)	46.998

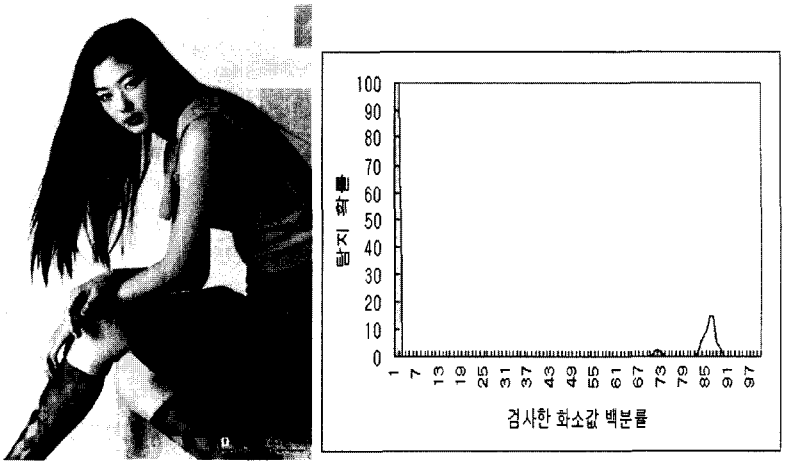
[표 5.1]은 실험에 사용된 영상의 원본과 은닉물 사이의 신호 대 잡음비(signal to noise ratio : $10\log_{10}\frac{\text{평균신호전력}}{\text{평균잡음전력}}$)를 나타낸다.

[그림 9]는 이론적인 최대 삽입량이 $\frac{1,795,254-54}{8}\times\left[1-\left(1-(\frac{1}{2})^{8}\right)^{648}\right]\approx$206KB이고 그에 해당하는 189KB의 메시지를 삽입했을 때 통계 테스트 결과를 보여준다.

Ⅵ. 결 론

본 논문에서 제안한 심층 암호 알고리즘은 블록 암호를 사용하여 원본 매개체에 일어날 수 있는 변화를 최소화하면서 충분한 양의 비밀 메시지를 삽입할 수 있도록 고안하였다. 이론적으로 증명되는 안전한 심층 암호를 실용화시킬 수 있도록 랜덤한 일방향 함수를 안전성이 검증된 블록 암호에 적용시킨 것이다.

기존의 최하위 비트 치환 방식의 심층 암호와 비



〔그림 9〕 189KB 메시지를 삽입한 은닉물(좌)의 통계 테스트 결과(우)

교했을 때 통계적 공격에 강인하면서도 거의 대등한 양의 메시지 삽입이 가능하였다. 또한 본 논문에서 제안하는 방식의 심층 암호는 기존의 심층 암호와는 달리 메시지 삽입시 메시지의 기밀성을 증가시키기 위해 또 다른 암호화 알고리즘이 없어도 메시지의 기밀성을 부여할 수 있으므로 비밀 메시지를 기존의 압축 프로그램으로 압축한 후에 삽입한다면 원본에 대한 변화를 보다 감소시킬 수 있을 것이다.

참 고 문 헌

[1] Cachin, C., "An Information-Theoretic Model for Steganography", in *Proceedings of the Second International Workshop on Information Hiding, vol. 1525 of Lecture Notes in Computer Science*, Springer, pp.306~318. 1998.

[2] Stefan Katzenbeisser, Fabien A.P. Petitcolas. "Defining Security in Steganographic Systems". in *ICASSP 2002.*

[3] Nicholas J. Hopper, John Langford, and Luis von Ahn Computer Science Department, Carnegie Mellon University, "Provably Secure Steganography", In *Proceedings of CRYPTO 2002.*

[4] Andreas Westfeld, Andreas Pfitzmann: "Attacks on Steganographic Systems". *Information Hiding* pp.61~76. 21, *1999.*

[5] Kahn, D., "The Codebreakers-The Story of Secret Writing", New York,USA: *Scribner*, 1996.

[6] Simmons, G. J., "The Prisoner's Problem and the Subliminal Channel", in Advances in Cryptology, *Proceedings of CRYPTO'83,* Plenum Press, pp.51~67,. 1984.

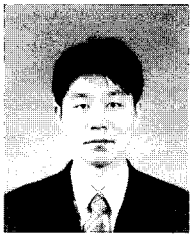
[7] Contraband, http://www.biol.rug.nl/hens/j/contrabd.exe

[8] U. Maurer, "A universal statistical test for random bit generators", in Advances in Cryptology - CRYPTO'90, A. J. Menezes and S. A. Vanstone, eds., vol. 537 of Lecture Notes in Computer Science, Springer-Verlag, 1991, pp.409~426.

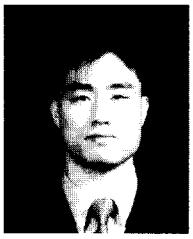
[9] Niels Provos, "Defending Against Statistical Steganalysis", in *Proceedings of the 10th USENIX Security Symposium,* pp.323~335, *2001.*

[10] R.Chandramouli and N.Menon, "Analysis of LSB based Image Steganography Techniques", Proceedings of ICIP 2001, Thessaloniki, Greece, October pp.7~10, 2001.

〈著者紹介〉



유 정 재 (Jeong-Jae Yu) 학생회원
1998년 8월 : 고려대학교 수학과 졸업
2001년 9월~현재 : 고려대학교 정보보호 대학원 석사 과정
<관심분야> 정보은닉이론, 디지털 워터마킹



김 종 현 (Jong-Hyun Kim)
2001년 2월 : 경희대학교 수학과 졸업
2001년 9월~현재 : 고려대학교 정보보호 대학원 석사 과정
<관심분야> 정보은닉이론, 디지털 워터마킹

박 종 혁 (Jong-Hyuk Park)
2001년 2월 : 순천향대학교 컴퓨터공학과 학사
2003년 2월 : 고려대학교 정보보호대학원 석사
2003년 3월~현재 : 한화S&C(주) 기술연구소 연구원
<관심분야> 정보은닉이론, 디지털컨텐츠 보안/관리

이 상 진 (Sang-Jin Lee)
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 2월 : 고려대학교 수학과 박사
1999년 3월~현재 : 고려대학교 정보보호대학원 부교수
<관심분야> 정보은닉이론, 블록 암호 및 스트림 암호의 분석과 설계, 암호 프로토콜

양 우 일 (Woo-Il Yang)
2003년 2월 : 고려대학교 정보보호 대학원 박사 수료
<관심분야> 정보은닉이론, 디지털 워터마킹