論文
3-5-8

부정확한 암호문의 복호화를 방지한 알고리즘 설계 (Design of Algorithm for Preventing Decryption of Inaccurate Ciphertext)

김 상 복* 구 명 모** 김 규 성***
(Sang-Bok Kim) (Myung-Mo Ku) (Gyu-Seong Kim)

요 약

본 논문에서는 잘못된 의미의 내용을 전달할 수 있는 문제점 해결을 위하여 부정확한 암호문의 복호화를 방지한 암호화 알고리즘을 설계하였다. 알고리즘은 복호화시 암호문의 에러를 체크하여 한 비트 이상의 에러가 포함되었을 경우 현재 진행중인 복호화를 중단함과 동시에 이전에 복호화 되어있는 복호문까지 모두 제거하여 수신자에게 아무런 내용도 전달하지 않게 된다.

실험은 비밀키의 대표적인 알고리즘인 DES알고리즘과 비교하였다. 실험결과에서와 같이 에러 비트가 포함되어 있는 부분은 원문과 다른 내용으로 복호화가 되었지만 100%의 복호률을 보였다. 제안한 알고리즘에서는 에러 체크를 통한 복호화로 에러 발생시 0%의 복호률을 나타내었다.

ABSTRACT

In this paper, we designed the encryption algorithm which protects ciphertext from deciphering solve the problem that it can pass the incorrect meaning. The algorithm checks the error of ciphertext during deciphering, and then if there are some errors of more than one bit. It stops the deciphering sequence and it doesn't pass any contents to receiver by removing the deciphering contents which was already done.

The experiment compared with DES algorithm that is representative algorithm of secret key. As the result of experiment, the part that error bit was included with was deciphered to one that was different from plaintext, but it showed 100% decipher_rate. by the algorithm showed decipher_rate of 0% with deciphering through the error check.

1. 서론

최근 급격한 네트워크의 발달로 인하여 정보화 보안에 관한 문제가 관건이 되고 있다. 네트워크 통신이 발전하면서 분산 네트워크의 이용과 더불어 상호간의 중요 정보를 교환하고 있다. 이로 인하여

* 정회원 : 경상대학교 컴퓨터과학과 교수
** 정회원 : 경상대학교 대학원
*** 정회원 : 공군 군전소 개발실

논문접수 : 2002. 4. 3
심사완료 : 2002. 4. 26

정보 보안문제가 큰 비중을 차지하고 있으며, 네트워크 상에서의 정보 보안을 위하여 암호화에 대한 연구가 계속적으로 필요한 실정이다.

정보 보호를 위하여 데이터의 암호화가 필요하다. 정보보호를 위한 암호화 방법에는 비밀키 암호화 방식과 공개키 암호화 방식이 있다.[1,2,3] 비밀키 암호화방식의 대표적인 알고리즘으로서는 DES(Data Encryption Standard)이다.[2,4] 이 방식은 64비트 키(key)(실제 56비트)를 적용하여 64비트의 평문을 64비트의 암호문으로 암호화시키는 대칭키 암호화 방법이다.[4,5,6] DES알고리즘에서는 대체와 치환이라는 2개의 기본적인 암호화 함수가 16회 반복적으로 적용한다는 특징이 있다.[2] 특히, DES는 공개키 암호화 방법에 비하여 암호화, 복호화시 계산량이 적어 암호 장치 구현이 용이한 장점이 있다.[3] 그러나 DES와 같은 공개키 암호화 방법은 송신자와 수신자가 동일한 하나의 비밀키를 가지고 있어 인터넷과 같은 개방형 시스템에서는 사용자들 간에 키의 안전한 사전분배에 따른 문제점으로 비밀키 암호화를 사용하기에는 어려움이 있으며, 네트워크를 통한 전송이나 제 3자로 인하여 암호문의 내용이 손상되었을 경우 그 내용이 원문과 다르지만 복호화가 이루어져 잘못된 의미가 전달될 수 있다는 가능성이 있다.[7,8]

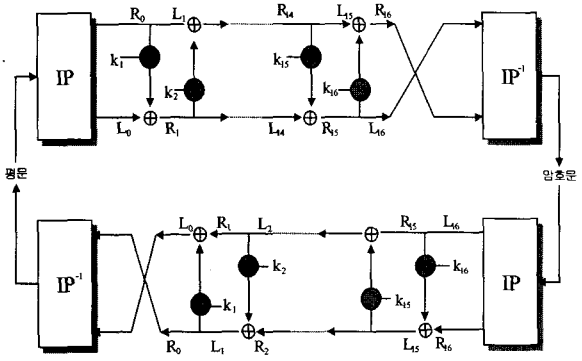
본 논문에서는 DES 암호화 알고리즘에서의 잘못된 암호문의 전송과 암호문의 내용이 손실되었을 경우나 네트워크 전송시 비트 오류가 발생했을 때는 손상된 내용에 관계없이 잘못된 암호문이 복호화되는 문제점을 개선하기 위하여 부정확한 암호문 복호화 방지 암호화 알고리즘을 설계하였다. 본 논문에서 설계한 암호화 알고리즘에서는 평문을 64비트 단위로 1차 암호화를 실행한다. 암호화에 이용된 키는 고정된 비트가 아닌 가변적인 비트를 이용하였다.[9] 1차 암호화된 암호문에 강도를 높이기 위해 1차 암호문과 128비트 키를 적용시켜 2차 암호화를 하였다. 암호화 과정에서 오류 체크 플래그 설정 과정을 행하여 정확하게 전달되지 않은 데이터 경우는 복호화를 할 수 없게 하여 제3자로부터의 암호문의 유출 및 손상으로부터의 보안적인 측면의 강도를 보다 높이도록 하였다. 암호화 부분은 64비트 크기의 한 블록을 대상으로 바이트 단위로 암호화를 진행하게 하였다. 각 블록을 바이트 단위로 암호화를 진행하여 각 블록은 각기 서로 다른 서브키(subkey)를 사용하여 마스크(mask) 시켜 암호문의 강도를 높이고자 하였다.

본 논문의 구성으로는 먼저 2장에서는 관련 연구인 DES 암호화에 대해서 살펴보고, 3장에서는 본 논문에서 제시한 부정확한 암호문 복호화 방지 암호화에 대해서 기술한다. 제4장에서는 본 논문의 암호화 알고리즘에 대한 실험 환경 및 실험 결과를 기술하고 마지막으로 5장에서 결론과 향후 연구 과제를 제시하였다.

2. 비밀키 암호화

비밀키 암호방식은 오래 전부터 널리 사용되어온 방식으로 송신측과 수신측에 똑 같은 키를 사용하여 암호화 및 복호화한다. 비밀키 암호방식의 대표적인 알고리즘으로는 DES암호화 알고리즘이 있다.

DES알고리즘은 평문을 64비트씩 평문 블록으로 나누고, 각각의 평문 블록 길이 56비트의 암호화 키로 16회 대치와 치환을 통해 64비트의 암호문으로 변환시키는 블록 암호방식이다.



[그림 1] DES알고리즘의 암호화/복호화

[Fig. 1] Encryption/Decryption of DES algorithm

[그림 1]은 DES알고리즘의 암호화와 복호화를 나타낸다. 그림에서 초기전치(IP)는 초기전치 규약표를 통하여 전치되어 암호화 과정을 수행하고 최종전치 규약표에 따른 최종전치(IP^{-1})를 처리함으로써 암호화

한다. 암호화과정에서 초기전치된 64비트는 32비트씩 좌우로 나누어져 좌측은 L_0 ~ L_{16}, 우측은 R_0 ~ R_{16}이 될 때까지 16단에 걸쳐 변환 과정을 반복한다. n번째 처리를 마친 L_n과 R_n은 (1), (2)로 표시한다.[1,2]

$$L_n = R_{n-1} \quad (1)$$

$$R_n = L_{n-1} + f(R_{n-1}, K_n) \quad (2)$$

암호문에서 평문을 복호화하는 과정도 평문으로부터 암호문을 생성시키는 과정과 같은 알고리즘을 이용한다. 즉 R_{16}, L_{16}을 입력하여 K_n을 n = 16에서 n = 1의 순으로 사용하면 암호화시키는 것과 동일한 알고리즘을 적용할 수 있다.

복호화 과정은 암호화 과정의 역 과정이므로 R_{n-1}과 L_{n-1}은 (3), (4)로 표시된다.[1,2]

$$R_{n-1} = L_n \quad (3)$$

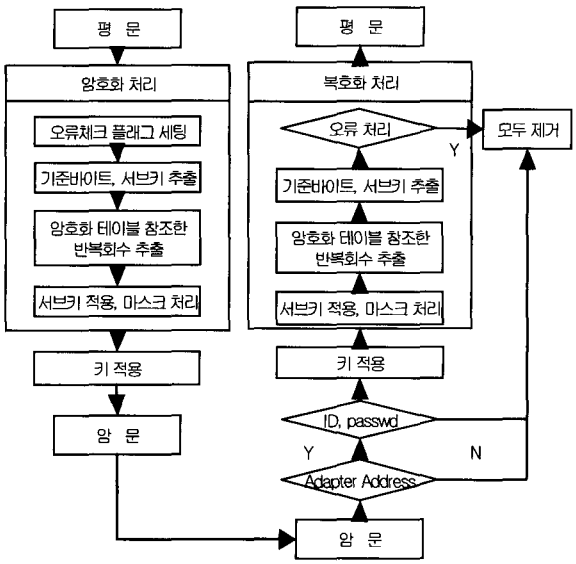
$$L_{n-1} = R_n + f(L_n, K_n) \quad (4)$$

따라서 암호화 과정에서 각 단에 사용되었던 동일한 키 K_n을 복호화 과정에도 다시 사용한다.

3. 부정확한 암호문의 복호화 방지 암호화

3.1 암호화/복호화 알고리즘 설계

본 논문에서 설계하는 암호화 알고리즘에서는 실시간 처리 방식인 대화식 메시지 혹은 파일 전송시 권한이 주어진 수신자만이 암호문을 복호화할 수 있도록 하였다. 분산 네트워크 환경에서는 전송되는 암호문이 완전하게 공개되는 것이나 다름이 없다. 이 때문에 제3자가 악의적으로 이 암호문을 입수하더라도 해독을 할 수 없도록 정보를 암호화하여 보호할 필요가 있기 때문에 [그림 2]와 같은 암호화 방법을 통하여 암호 강도를 높이도록 하였다.



[그림 2] 암호화/복호화 과정

[Fig. 2] Encryption/Decryption procession

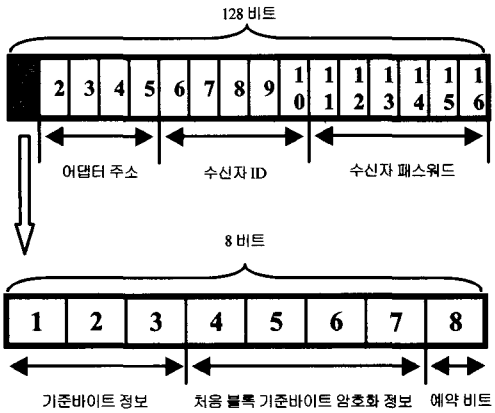
본 논문의 알고리즘에서는 암호화의 첫 번째 과정에서 오류체크 플래그를 설정하였다. 이는 전송도중에 발생할 수도 있는 암호문의 오류 및 악의적인 제 3 자로부터의 암호문의 탈취 및 수정으로 인한 잘못된 암호문의 오류를 조사하기 위한 것이다. 이때 오류체크 플래그 설정은 ASCII 코드로 문자를 입력할 경우 각 바이트는 최상위 한 비트를 사용하지 않는 점을 이용하여 본 논문에서는 이 비트 자리를 패리티 비트로 사용하였다.

오류체크 플래그 값은 홀수 패리티(odd parity) 방법을 이용하였다. 이것을 시작으로 암호화 과정을 통해 처리된 1차 암호문을 생성한다. 이어서 1차 암호문에 128비트 키 값을 64 비트크기로 마스크(mask)를 2번 더 행하여 2차 암호화를 행하여 암호화의 강도를 보다 높였다. 이 과정을 통하여 생성된 최종 암호문을 암호문 전송시스템을 통하여 수신자에게 전송한다. 암호문을 수신한 수신자는 수신된 암호문을 복호화하기 위해 자신의 컴퓨터 어댑터 주소를 입력한다. 어댑터 주소는 네트워크 어댑터의 물리적인 주소를 이용한다. 이 물리적인 주소가 정확한 경우 수신자는 자신의 ID와 패스워드를 입력하여 복호화를 위한 인증절차를 마치게 된다. 이 과정을 통하여 인증부분이 정확히 일치할 경우에는 복호

화 처리가 행해지고, 일치하지 않을 경우에는 수신한 데이터를 복호화되지 않도록 설계하였다. 이로서 부정확한 암호문의 복호화를 방지토록 하였다.

3.2 키(key) 구조

데이터가 완전 공개된 것이나 다름없는 분산 네트워크 상에서의 정보 전송과정에서 악의적인 제3자가 암호문을 입수하더라도 해독이 되지 않도록 할 뿐만 아니라 인가된 사용자에게도 정확한 의미의 암호문의 내용을 전달하기 위함에 있다. 특히 암호화 알고리즘 설계에 있어서 DES에서의 키 관리 등 효율성이 떨어지는 문제점과 DES의 오류 조사를 하지 않는 문제로 인하여 부정확한 내용이 전달 될 수 있는 단점을 해결하기 위하여 설계한다.



[그림 3] 키 구조
[Fig. 3] Key structure

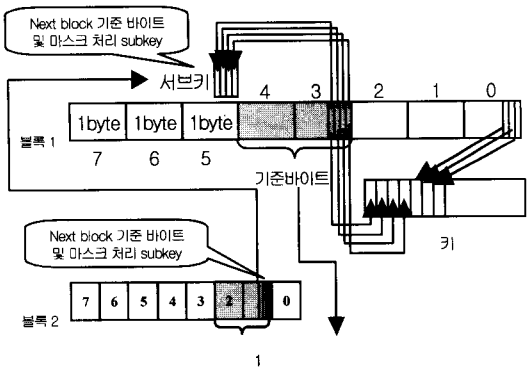
[그림 3]은 본 논문의 제안한 암호화에서 사용되는 키 구조에 대하여 나타내었다. 키 길이는 128비트의 길이를 가진다. 처음 1바이트에는 암호화되는 첫 번째 블록의 기준 바이트 정보 3비트와 암호화되는 첫 번째 블록의 기준 바이트에 대한 암호화 정보 4비트와 사용되지 않는 예약비트(1비트)가 저장된다. 키의 나머지 바이트들은 어댑터 주소(4바이트), 수신자 ID(5바이트), 수신자 패스워드(6바이트)가 저장된다. 특히, 어댑터 주소 값은 물리적인 어댑터 주소(12바이트)를 일정한 값과 마스크하여 4바이트로 압축하여 저장한다. 복호화는 이 값을 역 마스크하여 다시 12바이트 값으로 만든 후 비교하게 된다. 수신자 ID와 패스워드는 미리 확보된 정보를 이용하여 키에 저장한다.

3.3 암호화 알고리즘

[그림 4]는 키 생성과 기준 바이트 선택에 대하여 나타낸 그림이다. [그림 4]에서 블록1은 현재 암호화가 행해질 첫 번째 블록이며, 블록2는 블록 1의 암호화가 끝난 후 이어서 암호화가 이루어질 블록이다. [그림 5]는 키 생성과 기준 바이트 선택에 대하여 나타낸 그림이다. [그림 4]에서 블록1은 현재 암호화가 행해질 첫 번째 블록이며, 블록 2는 블록1의 암호화가 끝난 후 이어서 암호화가 이루어질 블록이다.

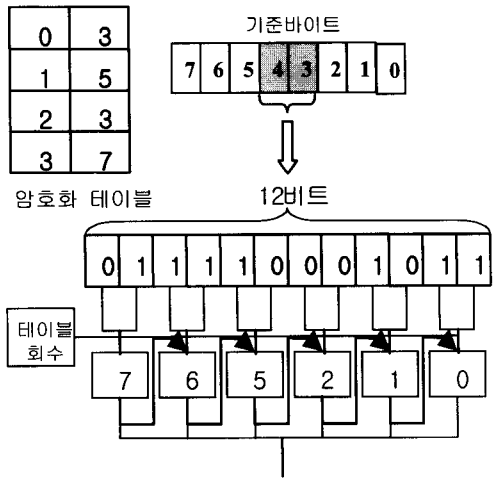
먼저 블록1에서 최하위 3비트를 추출하여 현재 암호화를 행하는 블록에서의 기준 바이트를 선택한다. 본 논문에서는 암호화가 진행될 첫 번째 블록에서 최하위 3비트를 추출하여 사용하는 것으로 설계했다. 추출된 3비트 정보를 이용하여 현재 암호화가 진행되는 블록 내에서의 기준 바이트를 선택하는데, 예를 들어 3비트에 "100"이라는 값을 가지고 있다고 가정하면, 이는 10진수로 4를 의미한다. 따라서 첫 번째 블록에서의 기준 바이트는 4번 바이트가 된다. 일단 기준 바이트가 정해지면 기준 바이트의 오른쪽 한 바이트를 묶어서 2바이트 크기를 기준 바이트로 정하게 된다. 그래서 [그림 5]에서 4번 바이트와 오른쪽에 있는 3번 바이트를 묶어서 2 바이트 크기의 기준 바이트가 형성된다.

본 논문에서는 암호화를 위하여 설정된 기준 바이트들의 비트열 16비트 중 최하위 4비트를 제외한 나머지 12 비트를 대상으로 2비트씩 하나의 그룹을 형성하여 각 그룹 당 암호화 알고리즘을 통하여 처리를 행하는 현재 블록의 각 바이트들의 포인터로 이용한다.



[그림 4] 블록 암호화 구조

[Fig. 4] Structure of block encryption



[그림 5] 암호화 처리

[Fig. 5] Encryption process

그러므로 기준 바이트를 제외한 나머지 6 개의 바이트들을 위해서 모두 12비트가 있어야 한다. 그래서 1바이트로는 이 모든 정보를 표현하기가 어렵기 때문에 2바이트를 기준 바이트로 선택한다. 이렇게 선택되어진 기준 바이트 16비트에서 최하위 4비트의 정보를 추출하여 키 값에 저장하고 이를 동시에 서브키(subkey)값에 저장한다. 이렇게 하는 이유는 저장한 4비트의 서브키는 다음 진행할 암호화 과정에서 사용하는 블록의 기준 바이트 선택 및 현재 진행하고 있는 기준 바이트의 마스크 암호화에 사용하기 위해서이다.

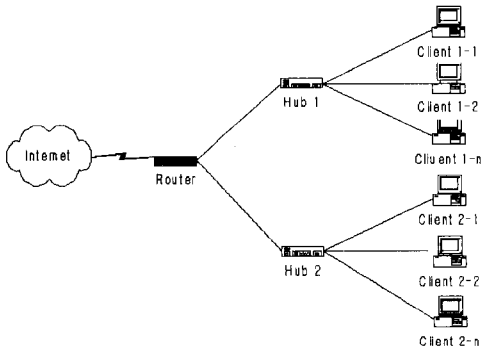
4. 실험

4.1 실험환경

본 논문에서 제시한 암호화 알고리즘을 바탕으로 한 실험환경은 펜티엄 PC(PII-350Mhz 3대), 운영체제는 윈도우즈 2000, 개발언어는 VC++ 6.0이다. 네트워크 환경은 10Mbps LAN상에서 구현하여 실험한다. 데이터 암호화 전송 및 복호화는 본 논문의 알고리즘을 적용하여 제작한 암호문 전송 시스템 시뮬레이터를 이용하였다.

먼저, 실험은 DES알고리즘과 제안한 알고리즘을 통한 암호화된 암호문을 여러 실험 조건을 주어 비교 분석한다. DES알고리즘에서 발생할 수 있는 문제점에 대한 부분을 실험을 통하여 알아보고, 제안한 암호화 알고리즘의 실험 결과를 통하여 DES에서의 문제점을 해결한 결과를 보인다.

[그림 6]은 암호문 전송 시스템의 시뮬레이션을 위한 네트워크 구성도에 대하여 나타내었다.

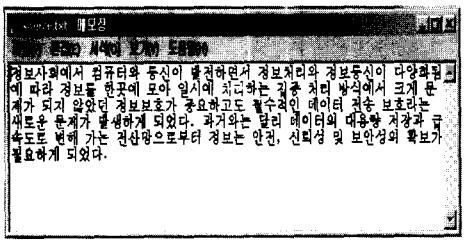


[그림 6] 실험을 위한 네트워크 구성도

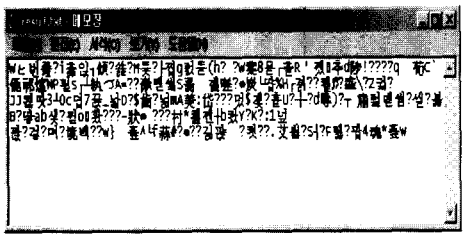
[Fig. 6] Network architecture for experiment

4.2 실험결과

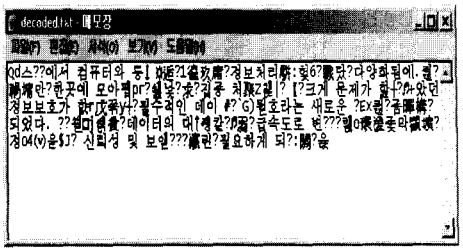
DES 암호화 알고리즘과 제안한 암호화 알고리즘을 통한 복호화 결과를 비교한다. 먼저 본 실험은 평문을 암호화시킨 암호문에 비트 에러를 적용하여 복호화한다. 실험 조건으로는 3블록 주기로 1비트의 에러를 적용한다. 비트의 에러 적용은 1비트 값을 평문과 상이한 값을 바꾸는 것으로 한다. 이렇게 에러 비트를 적용했을 경우 DES 암호화 알고리즘을 통하여 복호화한 결과와 본 논문의 알고리즘으로 처리한 복호화 내용을 비교하게 된다. [그림 7]은 평문을 나타내었다.



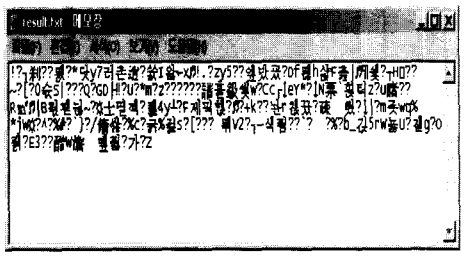
[그림 7] 평문
[Fig. 7] Plaintext



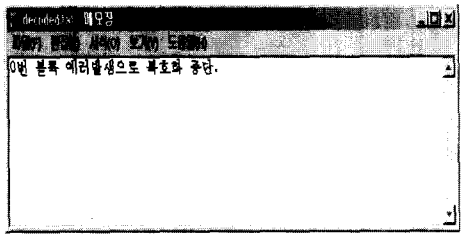
[그림 8] 암호문
[Fig. 8] Ciphertext



[그림 9] DES 알고리즘의 복호문
[Fig. 9] Deciphertext of DES algorithm



[그림 10] 제안 알고리즘 암호문
[Fig. 10] Ciphertext of proposed algorithm



[그림 11] 제안 알고리즘 복호문
[Fig. 11] Deciphertext of proposed algorithm

[그림 8]은 [그림 7]의 평문을 암호화시킨 후 홀수 블록마다 비트에러를 적용한 암호문에 대하여 나타내었다. [그림 8]에서처럼 암호화한 내용은 에러가 적용된 블록과 에러가 적용되지 않은 블록은 시각적으로 구분하기 힘들다. [그림 9]는 비트에러 적용된 암호문을 64비트 블록단위로 복호화한 내용에 대하여 나타내었다. 그림에서 복호화 내용을 살펴보면 평문과 같이 복호화된 부분과 복호화되지 않은

부분이 나타나게 된다. 결론적으로 평문과 같이 나타나지 않은 부분은 암호문에 비트에러를 적용한 부분을 의미한다.

DES 알고리즘에서 64비트 단위로 암호화를 시키는 블록 암호화 알고리즘이다. 연속적으로 64비트 문자열 암호화방식을 취하기 때문에 똑 같은 키가 적용되었을 경우 해독과정에서 에러부분이 포함되지 않은 블록의 글자가 정확히 복호화가 되는 것으로 볼 수 있다. 따라서 중요한 내용이 담긴 파일과 메시지의 경우에는 그 내용상 치명적일 수 도 있는 문제점을 나타내어 주고 있다. 이는 파일의 크기와 메시지의 길이와 손실정도의 크기에 따라 다르지만 암호화된 결과가 DES의 키 값만 알고 있는 경우에는 쉽게 해독이 될 수 있다는 결과이기도 하다.

또한 DES알고리즘에서는 지금까지의 결과와 같이 비트에 에러가 있는 부분은 평문과 같이 복호화가 되지 않았지만 복호화 과정에서 평문과 의미는 상이하지만 평문으로 잘 못 파악할 수 있는 내용이 나타날 수 있다. 즉, 내용 중에서 해독은 안되었지만 복호화 과정에서 글자처럼 조합이 되는 경우도 발생하는 경우이다. 즉, DES에서는 암호문에 비트의 에러 유무를 확인하지 않기 때문에 무조건 복호화를 시키게 된다. 이러한 방법을 통하여 잘못된 의미를 수신자에게 전달하면 치명적인 결과를 초래할 수 있다는 문제점을 나타내고 있다. [그림 10]은 DES와 마찬가지로 암호화될 같은 암호문에 대하여 나타내었다. 본 논문의 암호화 실험 조건은 DES 알고리즘을 통하여 실험한 조건과 같이 적용한다. [그림 11]은 제안 알고리즘 암호화 과정을 통하여 복호화를 시킨 결과를 나타내었다. 여기서는 수신자의 ID, 패스워드, 어댑터 주소의 개념을 포함하지 않는다. 그러나 암호문에 비트에러 유무만 확인하여 복호화를 시켰다. 복호화 결과 아무런 내용이 나타나지 않았다. 복호문에 나타나 있는 내용은 이미 실험에서 0번 블록에 비트에러를 적용하였기 때문에 복호화 시작과 동시에 복호화 과정을 중단하였다. 이때의 에러의 정보만 나타나낸 것이다. 지금까지의 실험 결과를 분석해 볼 때 DES알고리즘에서는 암호문의 에러 유무를 확인하는 과정이 없기 때문에 모든 암호문의 내용을 복호화하였다. 그러나 에러가 발생한 블록에 대해서는 평문의 내용과 같이 복호화되지는 않았지만 복호화 과정 중에 비트열의 조합으로 인한 잘못된 의미의 글자가 발생할 수 있는 문제점이 발생하였다. 본 논문에서 설계한 암호화 알고리즘에서는 결과와 같이 암호문의 내용에 한 비트라도 비트에러가 발생한 블록이 있다면 복호화를 하지 않았다. 이렇게 함으로써 잘못된 의미의 내용을 전달 할 수 있는 가능성을 없앴다.

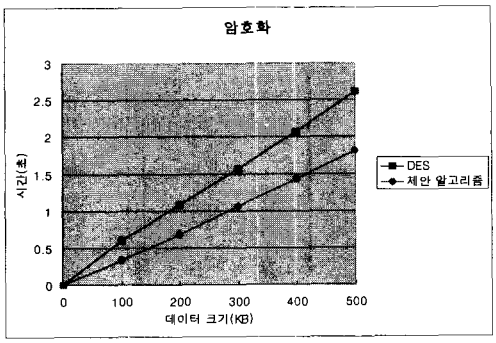
<표 1>은 DES와 본 논문의 제안 알고리즘에서 1KB문서를 1000번 반복한 복호률에 대한 실험 결과를 나타내었다.

<표 2> DES와 제안 알고리즘에서의 복호률

<Table 1> Decipher_rate in DES vs. proposed algorithm

비트에러	DES	제안 알고리즘
0비트	100%	100%
1비트	100%	0%
4비트	100%	0%
8비트	100%	0%

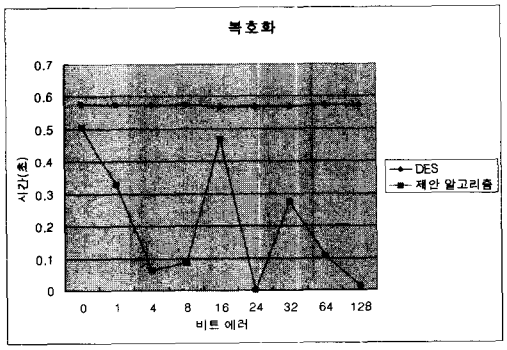
각 문서에 대하여 비트에러를 적용하였는데, 비트에 에러를 적용하지 않았을 경우(0비트)에는 DES와 제안 알고리즘 모두 100% 복호화가 이루어 졌다. 그러나 임의의 하나의 블록에 1비트, 4개의 블록에 대하여 각각 1비트씩 모두 4비트, 같은 방법으로 8비트에 에러를 적용했을 경우 DES는 모두 에러에 무관하게 모두 복호화가 되었다. 그러나 본 논문에서 설계한 제안 알고리즘에서는 한 비트라도 에러가 포함되어 있을 경우에는 복호화가 행해지지 않았다. 본 논문에서 설계한 제안 알고리즘에서는 네트워크 통신을 통하여 전송할 경우에는 모든 패킷을 정상적으로 수신했을 경우에만 복호화가 이루어진다는 것을 의미한다. 또한 오프라인(offline)에서도 암호화된 문서상의 내용이 손상이 발생했을 경우와 내용이 변경되었을 경우에도 복호화가 이루어지지 않는다는 장점이 있음을 확인하였다. [그림 12]는 DES와 제안 알고리즘의 암호화 수행시간을 비교한 그림을 나타내었다. 그림은 데이터의 크기에 따른 수행시간을 나타낸다.



[그림 12] 암호화 수행시간 비교

[Fig. 12] Encryption time of DES vs. proposed algorithm

암호화 때 데이터의 크기가 커짐에 따라 암호화의 수행 시간은 제안한 알고리즘이 다소 빠른 것으로 나타났다. DES에서의 이런 결과는 64비트의 블록을 대상으로 16회 라운드를 거치는 과정의 시간과, 키 스케줄(schedule)에 의해서 키 생성하는 수행하는 부분에서 비교적 많은 시간이 걸리게 된다는 것을 알 수 있다. 제안 알고리즘에서는 서브키 생성을 위하여 비트 연산을 하지 않으며, 64비트의 하나의 블록을 대상으로 라운드가 수행되지 않고 각 1바이트씩 암호화 테이블을 참조한 마스크 과정을 수행함으로써 시간이 단축되었다고 볼 수 있다.

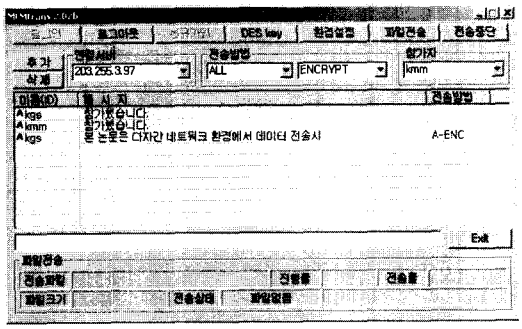


[그림 13] 복호화 수행시간 비교

[Fig. 13] Decryption time of DES vs. proposed algorithm

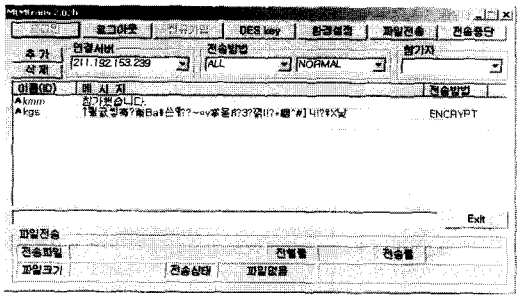
[그림 13]은 복호화에 걸리는 수행시간을 비교하였다. 그림의 결과는 임의의 블록에 1비트에러를 적용한 암호문을 복호화에 걸리는 시간대를 측정한 결과를 나타낸다. DES 암호화에서는 에러유무를 확인하지 않고 복호화(이전 실험 결과)하기 때문에 매 암호문마다 일정한 시간대를 형성한다. 그러나 제안 알고리즘에서는 에러 발생 여부를 조사하기 때문에 에러확인 시 복호화를 중단한다. 즉, 임의의 블록에 에러를 적용하였기 때문에 에러가 없는 블록에 대해서는 복호화를 수행한다. 그러나 에러를 만나면 즉시 중단과 동시에 이전의 모든 복호화된 내용을 지운다. 이런 결과를 바탕으로 [그림 13]과 같은 형태의 그래프를 형성하였다.

[그림 14, 15, 16, 17]은 본 논문의 시뮬레이션한 암호문 전송 시스템의 송, 수신자의 화면에 대하여 나타내었다. [그림 14]는 다자간 네트워크 환경에서 한 세션에 참가한 송신자의 화면을 나타내고 있다. 송신자는 수신자에게 그림에 있는 메시지를 암호화하여 실시간으로 수신자에게 전송한다. 여기서 송신자는 모든 수신자에게 전송하였다.



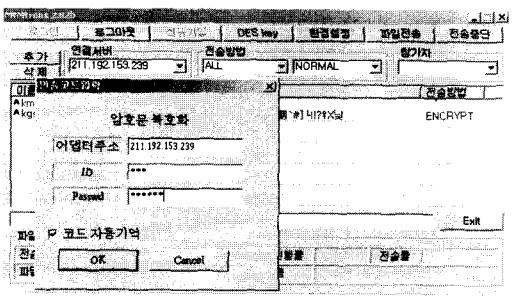
[그림 14] 송신자

[Fig. 14] Sender

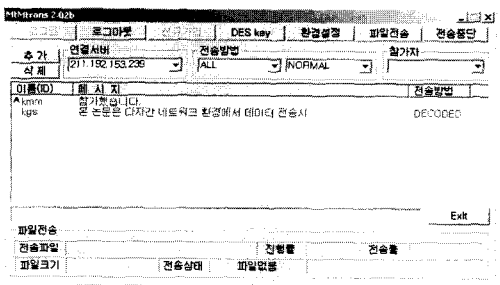


[그림 15] 수신자
[Fig. 15] Receiver

[그림 15]는 세션에 참가하고 있는 수신자의 화면을 나타내었다. 그림에서의 내용은 송신자로부터 전송 받은 암호문의 내용을 나타낸 것이다. 모든 수신자는 그림과 같은 내용의 메시지를 전달받을 것이다. 메시지를 수신한 수신자는 해독을 위하여 메시지를 선택하게 된다. [그림 16]은 복호화를 위하여 자신의 코드 값을 입력하는 화면을 나타내었다. 코드 값 확인을 위하여 어댑터 주소, 수신자의 ID, 패스워드를 입력한다. 이렇게 입력된 코드와 전달된 키 값을 비교하여 인가된 사용자인가를 확인하게 된다. 본 논문의 시뮬레이션에서는 어댑터 주소를 편의상 IP 주소를 부여하였다. 먼저 세션에 참가하고 있는 수신자의 정보는 서버에서 각기 다른 IP 주소와 ID, 패스워드를 관리하기 때문에 이 정보를 본 논문의 시뮬레이션에서는 이용하였기 때문이다. 이 코드 확인 과정이 끝나고 나면 비트에러 체크를 하게 된다. 에러의 확인이 없다고 확인되면 [그림 17]과 같이 암호문이 복호화된다.



[그림 16] 수신자의 코드 입력
[Fig. 16] Code input of receiver



[그림 17] 메시지 복호화
[Fig. 17] Message decryption

5. 결론 및 향후 연구과제

다자간 네트워크 전송에 있어서 기밀성을 요구하는 메시지는 인가된 사용자에게 정확한 메시지나 파일을 전송함을 목적으로 하고 있다. 따라서 본 논문에서는 인가된 사용자에게 정확한 메시지 및 파일을 전송하기 위하여 키 값에 어댑터 주소와 수신자의 ID 및 패스워드를 입력하여 암호화의 강도를 높이는 방법을 이용하였다.

DES는 실험결과에서와 같이 승인되지 않은 사용자들이 수신된 파일을 이용하여 복호화에 이용할 경우 DES키만 있으면 원문과 같이, 또는 손실정도가 심한 부분을 제외한 나머지 부분은 원문과 똑같이 복호화될 수 있는 문제점을 안고 있었다. 실험을 통하여 나타난 결과는 제 3자가 정상적인 방법으로 파일 및 메시지를 확보하여 암호화 및 복호화 키는 알고 있더라도 비밀 취급 등급과 어댑터 주소를 사용하여 신원확인을 받지 못하면 해독을 할 수 없었다. 인증이 되지 않은 사용자가 네트워크를 통하는 파일 및 메시지를 중간에 가로채었다면 DES는 키를 찾아서 해독할 수 있지만, 제안 알고리즘으로 처리된 메시지는 키를 가지고 있어도 어댑터 주소와 ID 및 패스워드를 알지 못하면 해독이 불가능하며, 또한 조금이라도 손상이 발생하였다면 암호화 테이블을 참조하지 못하고, 잘못된 정보를 가지고 있기 때문에 전혀 해독을 할 수 없는 결과를 얻을 수 있었다. 향후에는 현재 64비트의 블록 암호화를 탈피하여 암호화하는 평문의 보안등급에 따라 가변적인 블록 암호화 알고리즘으로 변경하여 메시지, 파일암호화 뿐

만 아니라 인증시스템에 적용될 수 있도록 연구되어야 할 것이다.

※ 참고문헌

[1] 이현열 譯者, "암호 조립법 입문" 대영사, 1997

[2] 박창섭 저, "암호 이론과 보안", 대영사, 1999

[3] 대한마이크로시스템(주) 세넥스테크놀러지스, "실무자를 위한 전산망 보안 세미나", 1999

[4] NBS, "Data Encryption Standard" FIPS pub, 46, u.s, National Bereaj of Standard, Washington DC, 1997.

[5] E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard,-Springer-Verlag, 1993.

[6] Copper smith, D. The Data Encryption Standard and its strength against attacks, IBM Research Report RC18613(81421), T.J. Watson Research Center. 1992.

[7] X. Lai and J. Massey, "A Proposal for a New Block Encryption Standard," Advances in Cryptology --- EUROCRYPT '90 Proceedings, Springer-Verlag, pp. 389--404. 1991.

[8] M.Matsui, "Linear cryptanalysis method for DES cipher, Advances in cryptolog- y-Eurocrypt '93, LNCS765, spring-verlag,pp.386-397, 1993.

[9] B. Schneier. Description of a new variable-length key, 64-bit block cipher (Blowfish). In R. Anderson, editor, Fast Software Encryption, volume 809 of Lecture Notes in Computer Science, pages 191--204, 1994. Springer Verlag.

김 상 복

중앙대학교 전자공학과 박사

현 경상대학교 컴퓨터과학과 교수

관심분야 : 멀티미디어 통신, 컴퓨터 네트워크, VHDL, 컴퓨터구조

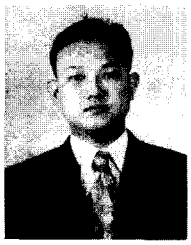
구 명 모

경상대학교 컴퓨터과학과 졸업 (석사)

경상대학교 컴퓨터과학과 박사 과정

관심분야 : 멀티미디어 통신, 컴퓨터 네트워크, 영상처리, 암호학

김 규 성



사후 90기

경상대학교 컴퓨터과학과 졸업 (박사)

공군 제3훈련비행단 정보통신대대 전산중대장

현 공군 군수전산소 개발실 사무자동화 반장

관심분야 : 컴퓨터 네트워크 보안, 암호학