키 길이 증가를 위한 확장 DES에 관한 연구

이 상 번, 남 길 현
국 방 대 학 원

A Study of Expanded DES for Improving Key Length

Sang-Beon Lee, Kil-Hyun Nam
National Defense College

요 약

관용키 암호시스템에서 가장 널리 실용화되어 사용되고 있는 알고리즘은 DES(Data Encryption Standard)이지만, DES의 보안성에 대해서는 공표이래 많은 비판과 논란이 있었으며 그중 56비트의 키로서는 충분한 비도를 제공하기에는 너무나 짧다는 것이 하나의 비판이다.

본 논문에서는 총 112비트의 키가 사용되도록 128비트의 평문을 네 부분으로 나누어 좌우 두개씩 DES알고리즘에 적용하고 좌우측을 교환하는 확장 DES를 제안하고 이를 분석하였다.

I. 서 론

최근 개인이나 조직 또는 국가의 중요한 정보들이 컴퓨터와 각종 전산망를 이용하여 신속히 저장되고 처리됨으로써 데이타들을 안전하게 보관하고 전송하기 위한 데이타 보안문제는 매우 중요한 문제로 인식되고 있다.

데이타 보안에는 여러가지 방법이 있지만 그 중 데이타를 암호화하는 방법은 데이타의 형태를 변형함으로써 비밀성을 보장할수 있고, 데이타의 내용을 읽지 못하게 함으로써 일반적인 수정을 거부할수 있기 때문에 가장 안전한 데이타 보안방법으로 사용되고 있다.

데이타를 암호화하는 관용키 암호시스템 중 가장 보편적으로 널리 실용화되어 사용되고 있는 알고리즘은 70년대 초 미국 정부 NBS(National Bureau of Standards)가 공개모집을 통해서 개발한 DES(Data Encryption Standard)[1]이다.

이 DES는 70년대 말부터 컴퓨터통신망의 확대와 더불어 급증한 보안사고로 인하여 데이타를 필히 암호화하여야 한다는 각별한 요구와 비교적 간단하게 구현하여 사용할 수 있다는 장점으로 미 정부 및 국가표준으로 채택되어 그 사용폭을 급속히 넓혀 왔으며 국제표준으로도 인정되어 사용되고 있으나, DES는 공표이래로 암호강도에 대한 많은 논란과 공개비판을 받아 왔는데 그 대부분은 DES에서 사용하고 있는 키(key) 길이가 요구되는 암호강도에 비해 너무 작다는 것과 암호 결과를 쉽게 풀수 있도록 하는 어떤 비밀방안(trapdoor)이 포함되지 않았느냐 하는 것이다[2,3,4].

또한 1985년 후반 NSA(National Security Agency)는 1988년 12월 이후부터는 더 이상 DES를 지원하지 않겠다는 뜻을 공식적으로 발표하였으나[5], 그 실용성때문에 상용으로 지금까지도 널리 사용되고 있다. 따라서 DES의 약점을 보완하여 DES의 생명을 보다 길게 하고자 하는 연구와 DES를 대체하여 쓸 수있는 새로운 암호알고리즘 개발이 다양하게 연구 및 개발되고 있는 실정이다.

본 논문에서는 DES의 여러가지 분석과 비판들을 고찰하고, DES의 가장 강한 비판이 되고 있는 키 길이를 증가하여 비도를 높일수 있는 DES의 확장방안를 제안하고 이러한 확장 방안을 지금까지 연구 발표된 DES의 분석방법 일부를 적용하여 평가하도록 한다.

Ⅱ. DES(Data Encryption Standard)

DES는 64비트의 평문을 56비트의 키를 사용하여 암호화하는 알고리즘으로서 기본구조는 평문의 각 비트 순서를 바꾸기위한 초기 재배열 부분, 암호화 키를 포함하여 재배열과 치환으로 이루어진 암호화 과정을 16회 반복 수행하는 부분 및 각 비트 순서를 다시 초기 재배열의 역배열로 바꾸는 최종 재배열 부분으로 이루어 진다.

DES는 64비트를 좌우 32비트로 나누어 암호화과정을 16회 반복하는데 1회의 암호화 과정을 수식으로 표현하면 다음과 같다.

$$L_i = R_{i-1} \quad (1)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \quad (2)$$

L : 왼쪽 32비트　　R : 오른쪽 32비트

⊕ : XOR연산　　i : 라운드 수

(1),(2)식을 간단히 변형하면 다음과 같다.

(1)식에서 $R_{i-1} = L_i$ (3)

(2)식에서 XOR연산의 특성을 이용하면

$$L_{i-1} = R_i \oplus f(R_{i-1}, K_i)$$

$$= R_i \oplus f(L_i, K_i) \quad (4)$$

(3),(4)식을 살펴보면 i라운드의 결과로서 i-1라운드의 결과를 유도할수 있다.

DES의 암호함수 f에서는 입력되는 32비트 중 일부 비트를 반복 사용하여 48비트로 확장하여 재배열하고 48비트 키와 XOR연산을 한다음 6비트씩 8개의 S(Substitution)-box를 통해 각각 4비트 값으로 치환시키고 이를 다시 재배열시키는 작업을 하게 된다.

DES의 키 스케쥴은 48비트의 서브키(subkey)를 생성하여 각 라운드별로 사용하게 한다. 키 스케쥴은 최초 사용자가 제공하는 64비트의 키에서 8개 비트를 제거하고 남은 56비트를 처음 각 비트의 위치를 바꾸어 재배열하고 28비트씩 양분한 다음 이를 각 라운드 별로 1 - 2비트씩 circular left shift시키어 각각의 28비트에서 24비트를 선택하여 재배열하고 이 때 생성된 48비트를 라운드별 서브키로서 사용한다.

Ⅲ. DES의 비판과 확장 방안 제안

1. DES에 대한 비판과 논란

DES의 기본구조중 초기 및 최종 재배열은 공개된 암호알고리즘으로서는 비도에 아무런 도움을 주지 못하며 오히려 DES를 소프트웨어로 구현시 수행속도만 20%정도 증가시키는 단점을 지니고 있다[2].

DES에 대한 가장 중요한 논란이 되어 왔던 것들은 DES설계자가 S-box, P-box 및 키 스케쥴에 대한 설계시 고려사항들을 발표하지 않은 것과 16라운드의 DES가 충분하느냐 하는 것, 그리고 56비트의 키가 충분한 비도를 제공할수 있는가 하는 것이다[6].

첫번째의 논란은 DES설계자 자신들은 암호문의 복호화를 쉽게할 수 있도록 DES의 내부에 어떤 비밀방안을 숨기지 않았느냐 하는 의심을 갖고 지금까지 많은 연구가들이 DES에 대한 연구를 하였지만 아무도 비밀방안을 찾아내지는 못하였다.

두번째의 논란인 16라운드에 대한 의심은 8라운드면 모든 발견 가능한 연관들이 제거됨으로써 16라운드는 충분한 암호알고리즘이 된다고 분석되었다[7].

세번째의 논란인 56비트의 키에 대해서는 충분한 비도를 제공하지 못한다는 비판이 계속되었고 이를 보완하고자 하는 방법들이 연구되어 왔다.

2. DES에 대한 다양한 공격방법

● exhaustive 공격방법

일반적으로 exhaustive 공격(혹은 brute force 기법)이란 직접적인 탐색방법을 통하여 키나 평문을 구하는 방법으로 키에 대한 공격 시는 가능한 키 전부를 사용하여 평문을 암호화하거나 암호문을 복호화하여 일치여부로서 키를 찾아내는 방법이다.

Hellman 과 Diffie[2]는 DES의 56비트 키로서는 exhaustive 공격에 약하다고 주장하였다. Known-plaintext 공격방법을 적용하면 암호화에 가능한 2^{56} = 약 10^{17}개의 키로서는 한개의 키를 암호화하는데 1μs가 소요된다고 할 때 약 10^{6}일이 필요하므로, 백

만개의 DES칩을 연결하여 병렬처리할수 있는 특수한 기계를 제작하여 이용하면 하루가 소요되고 평균 12시간 내에 키를 발견할 수 있다고 주장한 것이다. 따라서 칩제작 및 컴퓨터 기술발전 속도을 고려할 때 10년내에 DES는 그 생명력을 잃을것이며 Known-plaintext 공격에 보다 강해지기 위해서는 키 길이를 늘리는 것이 최선의 방법이라고 주장하였다.

● time-memory tradeoff 공격방법

Hellman은 컴퓨터의 기억용량이 더 많이 요구되고, 사전계산시간(precomputing time)이 많이 요구되지만 exhaustive 공격보다 키 탐색시간이 덜 걸리는 time-memory tradeoff 방법을 제시하였다[8].

Hellman은 time-memory tradeoff 공격방법으로서는 메모리의 크기 및 탐색시간이 $N^{2/3}$정도 필요하게 되어 exhaustive 공격보다 효과적인 공격방법이 될 수 있다고 주장하였다.

그러나 이러한 공격방법을 위해서는 테이블을 형성하기 위한 수년간의 사전계산시간이 요구되는 점과, 또한 Known-plaintext의 chaining기법을 사용한 암호문에는 적용불가하다는 단점도 있어서 실제 적용상에는 문제점이 있다고 볼 수 있다.

따라서 Hellman은 time-memory tradeoff의 공격방법을 막기 위해서는 chaining기법의 사용이나 충분한 키길이를 사용할 것을 추천하고 있다.

● differential cryptanalysis 공격방법

E. Biham과 A. Shamir는 새로운 DES-like암호시스템의 공격방법인 differential cryptanalysis을 제시하였다[9]. 이 공격방법은 DES의 형태와 같이 암호화 과정을 반복하는 반복 암호시스템(iterated cryptosystem)에 적용되는 공격형태로서, 다수의 특정 평문쌍과 이에 대응하는 암호문쌍에 대하여 통계적인 방법을 적용하여 사용된 키를 구하는 방법이다.

differential cryptanalysis를 적용하여 일정 라운드의 DES를 공격하는 방법은 최종 라운드의 암호함수에 적용된 입력 XOR과 출력 XOR를 알아냄으로써 최종 라운드의 암호함수에 적용된 키를 찾아 내는 방법을 사용한다.

differential cryptanalysis에서는 n라운드의 특성을 정의하고 이 특성들을 연결하여 특정한 입력 XOR이 마지막 라운드에 적용되도록 한다. 여기에서 특성이라 함은 특별한 입력 XOR, 가능한 출력 XOR, 필요한 중간 XOR 및 이러한 값이 적용될수 있는 확율을 말한다.

differential cryptanalysis에서는 확율로서 정의된 n라운드의 특성을 사용함으로써 공격자는 정의된 것과 동일한 출력 XOR이 나올 때까지 다수의 입력 XOR들을 적용하여야 되고 특성과 맞는 입력과 출력이 구해질 때에 이들 쌍이 키를 구할 수 있는 올바른 입출력 쌍이 되며, 또한 짧은 특성들을 연결해서 보다 긴 라운드의 DES를 공격할때에는 긴 라운드에서의 확율은 사용된 특성들 확율의 곱으로써 나타낼수 있으므로 키를

찾기 위한 작업량은 증가하게 된다. 따라서 differential cryptanalysis는 어느 일정한 라운드까지는 exhaustive 공격방법보다 매우 빠르나 일정 라운드 이상에서는 느리게 된다.

3. DES를 이용한 비도 증가 방안

● DES를 가장 안전한 방법으로 사용하는 방법

일반적인 암호시스템에서와 마찬가지로 DES를 안전하게 사용하기 위해서는 키 선택이 아주 중요하다. Weak key나 Semi-weak key는 키 선택시 반드시 배제해야 되며, 일정한 틀에 의한 키 선택보다는 랜덤하게 키를 선택함으로써 안전한 키의 사용을 보장할수 있다. 또한 키를 대문자 알파벳만을 이용하는 등 제한된 범위내에서 선택할 경우는 효과적인 키 길이를 줄이는 결과를 초래하게 되므로 키 선택은 확실하고 안전성 있는 키생성 알고리즘를 통해서 이루어 져야 할 것이다.

키 선택뿐만 아니라 키 분배에서도 안전한 시스템을 사용하여야 한다. 일반적으로 키 분배에서 가장 좋은 방법은 공개키(public key)암호시스템을 이용하는 것이나, 다만 공개키 암호시스템의 단점인 속도가 느리다는 점을 배제하기 위해서는 사용될 소수(prime number) 등을 사전에 계산하여 테이블로 작성해 놓고 랜덤하게 선택해서 사용할수도 있다.

키의 사용기간에 대해서는 동일한 키를 장시간 사용하기 보다는 비교적 짧은 기간마다 교체하는 것이 좋으며, 1회용(one-time session)키를 사용하거나 먼저의 키가 다음의 키에 영향을 주어 키가 계속 변경되도록 하는 autokey(running key)를 적용하면 매 암호화마다 다른 키를 사용하게 되므로 키의 보안성을 증가시키는 효과를 볼 수 있다.

DES는 평문을 64비트 즉 8바이트단위로 암호화하기 때문에 8바이트씩 반복되는 평문을 암호문에서도 그대로 반복되므로 미리 평문을 압축하여 불필요한 것(redundancy)을 제거하여 암호화하거나 안전한 운용모드를 이용하여 암호화하는 것이 좋다.

DES의 운영모드에는 CBC(cipher block chaining)모드, CFB(cipher feed-back)모드 및 OBFB(output-block feedback)모드등이 있다[10].

● DES의 다중사용

이중암호화방법은 평문을 다른 키로 두번 중복하여 암호화하는 것이다. 이러한 방법은 키의 크기를 112비트로 늘리는 효과를 가져와 암호공격에 강하게 할 수 있다고 단순히 생각될 수 있다. 그러나 closed 문제와 meet in the middle 공격의 문제로 인하여 이중암호화방법의 암호강도는 2^{112}에 훨씬 미치지 못하는 것으로 평가되고 있다[2,4].

삼중암호화방법은 이중암호화와 같이 두개의 키를 이용하거나 또는 각기 틀린 세개의 키를 이용하여 세번 중복 암호화하는 방법이다[11]. 두개의 키를 이용한 삼중암호화방법은 이중암호화에 대한 meet in the middle 공격을 피할수 있는 방법으로 생각되

나, Chosen-plaintext 공격에서는 이중암호화와 동일한 메모리용량이나 연산정도를 필요로 하게 되어 이중암호화와 비슷한 비도밖에 되지 않는 것으로 평가된다.

각기 틀린 세개의 키로 암호화하는 삼중암호화방법은 meet in the middle 공격때문에 $(2^{56})^2 = 2^{112}$의 암호강도를 갖게 되는 것으로 판단되나[12], 암호화시 속도문제와 많은 키를 갖고 있어야 하는 면이 단점으로 지적될수 있을 것이다.

● DES알고리즘의 부분수정

DES의 소프트웨어를 일부 수정해서 암호공격에 강하도록 하는 개선방법도 다양하게 생각할수 있다[13]. 이러한 개선방법은 DES내부에 숨어 있을지도 모를 비밀사항을 방지하기 위하여 S-box나 키스케쥴을 랜덤하게 운영하는 방안이 있을 수 있다. 한 예로서 키에 따라서 S-box 테이블을 새로 작성하여 사용하든가 키에 의존한 랜덤넘버(random number)를 발생하여 키 스케쥴의 비트 로테이션(rotation)으로서 사용하면 랜덤성을 제고할 수 있을 것이다.

단 이러한 수정이 DES의 암호강도를 급격히 낮출 수도 있으므로 신중한 판단이 요구되며[9], 하드웨어의 구현이 힘들기 때문에 보다 효과적인 방법이 요구된다.

4. 키 길이 증가를 위한 확장 DES의 제안

키 길이 증가를 위한 알고리즘의 확장 방안의 기본구상은 두개의 DES알고리즘을 합쳐놓아 총 112비트의 키가 요구되도록 하는 것이다. 즉 128비트의 평문을 네개의 32비트로 나누어 좌우 두개를 DES알고리즘에 적용하고 좌우측의 결과를 교환함으로써 키와 평문이 암호문 전체에 영향을 미치도록 하는 것이다.

두개의 DES 암호알고리즘을 보다 효율적으로 묶기 위한 방법으로 DES의 암호알고리즘 식을 확장적용하여 보자. 즉 네개의 32비트 평문을 각각 A, B, C, D라고 하면 DES의 암호알고리즘을 아래와 같이 변형하여 적용할 수 있다.

$$D_i = A_{i-1} \oplus f(B_{i-1}, K_i) \quad (5)$$

$$B_i = C_{i-1} \oplus f(D_{i-1}, K_i) \quad (6)$$

$$A_i = B_{i-1} \quad (7)$$

$$C_i = D_{i-1} \quad (8)$$

위의 식을 좌우를 바꾸어 다시 정리하면 다음과 같다.

$$B_{i-1} = A_i \quad (9)$$

$$D_{i-1} = C_i \quad (10)$$

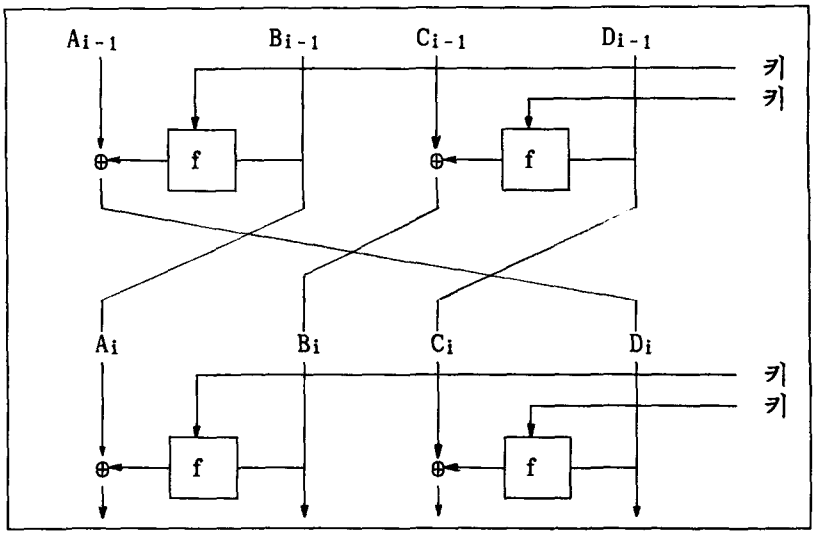
$$A_{i-1} = D_i \oplus f(B_{i-1}, K_i) = D_i \oplus f(A_i, K_i) \quad (11)$$

$$C_{i-1} = B_i \oplus f(D_{i-1}, K_i) = C_i \oplus f(C_i, K_i) \quad (12)$$

위의 (9)-(12)식으로서 i 라운드의 값으로 i-1 라운드의 값이 유도됨으로, DES알고

리즘을 확장적용시킬 수 있음을 알 수 있다.

이와같은 방법은 암호화과정 즉 식 (5) - (8)을 <그림 1>과 같이 도식할 수 있다.



<그림 1> 확장 DES의 암호화구조

좌우를 교환하여 DES를 확장 적용하는 방법은 여러가지로 생각할 수 있으나 라운드가 증가해도 암호함수가 적용되지 않는 부분이 존재하든가 아니면 하나의 평문 비트나 키 비트가 수정되었다 하더라도 생성된 암호문에 그 확산효과가 제한되는 구조적인 단점을 가지는 방법은 암호알고리즘에는 부적절하게 된다.

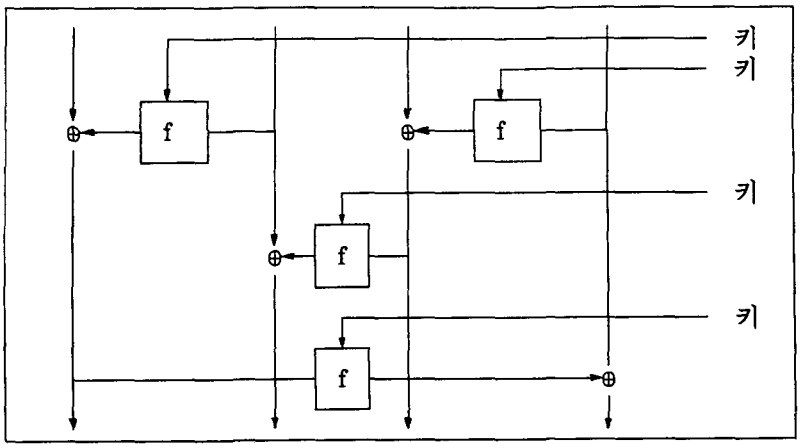
또한 라운드마다 좌우 64비트중 암호함수를 적용한 32비트와 암호함수를 적용하지 않은 32비트에 XOR연산을 수행하는 방법도 생각할 수 있으나 XOR연산은 하나의 비트가 오직 한 비트에만 영향을 주게 되어, 평문에 하나의 비트만 수정될 경우에도 전체 암호문이 바뀌도록 하는 일반적인 암호알고리즘에서 요구되는 암호 확산효과의 기대에는 못 미친다는 것이 커다란 단점이 된다.

따라서 가장 바람직한 확대 적용방법은 <그림 1>과 같이 라운드가 증가할수록 평문이나 키전체가 암호문에 영향을 주는 방법이라고 볼 수 있고, 또한 <그림 1>은 <그림 2>와 같이 변형해서 도식화하여 표현할수 있으므로 이는 단지 XOR연산 뿐만 아니라 XOR연산후 이 값이 암호함수와 연결된 형태가 됨으로 키 및 평문의 암호문에 대한 확산효과를 보강할수 있는 모형이라고 판단된다. <그림 1>과 같은 방안에 현 DES의 키 스케쥴을 적용하면, 확장 DES의 키 스케쥴과 암·복호화과정에 대한 전체 구조가 된다.

Ⅳ. 확장 DES의 분석

1. Avalanche effect

확장 DES에 대하여 암호화에 대한 일반적인 효과를 분석하기 위하여 DES특성인 Avalanche effect를 적용하여 분석하면 가장 쉽게 암호화의 효과를 가시적으로 판단할수



<그림 2> <그림 1>의 다른 표현

있다. 단, 많은 양의 평문과 키에 대한 적용은 실제 적용 상의 제약이 있기 때문에 특정한 평문 및 키만을 선택하여 암호화를 하고 그 결과를 분석한 결과 동일한 키에 대해서 평문이 한 비트만 바뀌더라도 암호문 전체가 변형되어 출력되었고 동일한 평문에 한 비트씩 차이가 나는 키를 적용해도 암호문은 전체가 변형되어 출력되었다.

2. 키 선택 범위

확장 DES알고리즘의 키는 좌우 동일한 키 스케쥴을 사용하기 때문에, 만약 키가 좌우 56비트의 값이 동일할 경우에는 라운드별 키값이 동일하게 제공되므로 128비트 평문도 좌우 64비트씩 동일하면 생성된 암호문도 좌우가 동일한 값으로 된다.

물론 이 경우도 생성된 암호문은 현 DES를 수행하여 얻은 64비트의 암호문과 동일한 비도를 제공하지만 생성된 암호문을 보면 평문의 좌우가 같다는 단서를 제공한다는 단점이 있기 때문에 이를 근절하기 위해서는 키 선택시 좌우가 동일하지 않도록 선택해야 된다. 이는 키선택의 폭이 그만큼 좁아진다는 단점을 배제할수는 없으나 그 량은 전체 키 선택의 갯수에 비해 상대적으로 매우 작기 때문에 크게 염려할 필요는 없다고 판단된다.

또한 확장 DES는 기존 DES의 키 스케쥴을 그대로 사용하기 때문에 DES의 Weak key나 Semi-weak key가 존재하게 된다.

3. 평문 및 키와 암호문에 대한 상호의존도

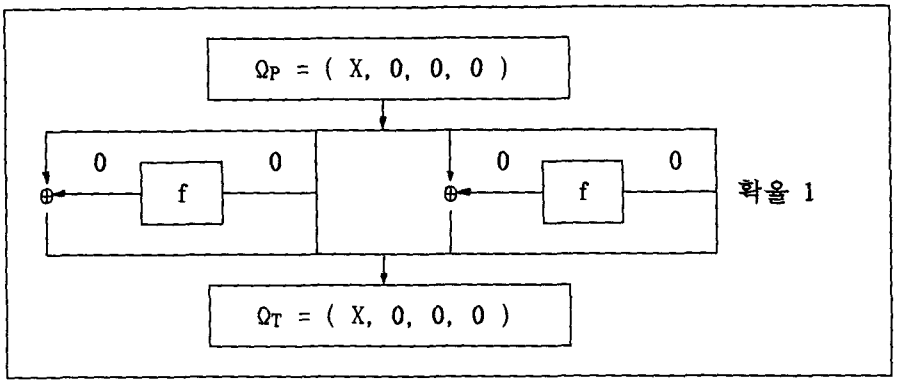
기존의 DES가 갖고 있는 재배열 및 치환 테이블을 확장 DES는 동일하게 사용하므로 확장 DES의 평문 및 키와 암호문 간의 상호의존도는 Meyer의 분석방법[14]이 바로 적용가능함을 알 수 있다.

따라서 Meyer의 분석방법을 적용하면 확장 DES는 7라운드 이후에 평문과 암호문의 100%의 상호의존도가 유지되고, 키와 암호문의 상호의존도는 6라운드 이후에 100%가

가능함을 알수 있다.

4. differential cryptanalysis에 의한 분석

Chosen-plaintext 공격방법인 differential cryptanalysis를 적용하여 확장 DES의 1라운드 특성을 <그림 3>과 같이 정의할수 있다.



ΩP : 입력 XOR ΩT : 출력 XOR
<그림 3> 확장 DES의 1라운드 특성

<그림 3>의 특성을 이용하면 확장 DES는 DES의 differential cryptanalysis 공격방법을 좌우측 두번을 하게 되면 전체 키값을 구할 수 있다는 논리로 확대할 수 있다. 그러므로 제안한 확장 DES는 differential cryptanalysis하에서는 키 길이가 늘어난 만큼의 비도증가를 예상할수 없기 때문에 이를 막을 수 있는 방법이 요구된다.

5. 기타 분석

암호공격기법중 exhaustive와 같은 공격방법을 사용하여 확장 DES의 112비트 키를 공격하기에는 너무나 많은 시간이 요구되므로, 보다 짧은 공격을 위한 short-cut한 방법이 발견되지 않은 한 확장 DES에 이러한 방법의 공격은 사용하기가 곤란하다.

구현된 확장 DES의 처리속도 문제는 평문을 128비트씩 암호화하므로 두개의 암호함수를 사용한다 해도 기존 DES와는 속도차이가 거의 없다고 볼 수 있으며, 더욱 비도에 도움을 주지 못하는 초기 재배열과 최종 재배열이 제거되어 소프트웨어로 구현시 속도는 더욱 빨라질 수 있다. 또한 현 컴퓨터 기술수준으로서는 확장 DES를 하나의 칩으로 구현가능하다고 판단된다.

V. 결 론

확장 DES는 키 길이의 증가로 exhaustive와 같은 Known-plaintext 공격방법에는 강할 것으로 기대되나 Chosen-plaintext 공격방법을 사용하는 differential cryptanalysis 공격방법에 대해서는 늘어난 키 길이 만큼의 커다란 효과를 기대하기는 어려운 것

으로 분석된다. 이러한 원인은 확장 DES가 differential cryptanalysis에 강하지 못한 구조적 단점으로 인한 것으로 판단되나 실제 우리의 실정에서는 Chosen-plaintext 공격방법은 적용되기가 매우 어려운 공격방법이기 때문에 본 논문에서 제안한 확장 DES방법은 새로운 관용키 암호알고리즘의 개발에도 기여할 것으로 생각되고 더욱이 키 생성 알고리즘과 일부 단점이 드러난 부분을 보완하면 바로 실용화될 수 있다고 판단된다.

< 참 고 문 헌 >

[1] NBS, "Data Encryption Standard," FIPS Pub. 46, U.S. Nationl Bureau of Standards, Washington, DC, Jan. 1977.

[2] W.Diffie and M.E.Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," IEEE, Vol.10, No. 6, pp. 74-84, June 1977.

[3] T.Goeltz, " Why not DES ?," Computer & Security, No. 5, pp. 24-27, 1986.

[4] M.E.Hellman, "DES will be totally insecure within ten years," IEEE Spectrum, Vol. 16, No. 7, pp. 32-39, July 1979.

[5] H.Joseph, "Demise of DES," Computer & Security, No. 5, 1986.

[6] C.P.Pfleeger, Security in Computing, Prentice Hall, 1989.

[7] A.Konheim, Cryptography : A Primer, John Willey & Sons, 1981.

[8] M.E.Hellman, "A Cryptanalytic Time-Memory Tradeoff," IEEE Trans. of Info. theory, Vol. IT-26, No. 4, pp. 401-406, July 1980.

[9] E.Biham and A.Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Abstracts of CRYPTO '90.

[10] C.R.Abbruscato, "Data Encryption Equipment," IEEE Com. Magazine, Vol.22, No. 9, Sep. 1984.

[11] R.C.Merkle and M.E.Hellman, "On the Security of Multiple Encryption," Comm. of ACM, Vol. 24, No. 7, pp. 465-467, July 1981.

[12] D.Coppersmith, "Cryptography," IBM J. Res. Develop., Vol. 31, No. 2, pp.244-248, Mar. 1987.

[13] J.M.Carroll, "Strategies for Extending the Useful Lifetime of DES," Computer & Securuty, No. 4, pp. 300-313, 1987.

[14] C.Meyer and S.Matyas, Cryptography, John Wiley & Sons, 1982.