블록 암호알고리즘을 위한, 추적불가능한 동적 키를 갖는 연산모드

김윤정○ 조유근
서울대학교 컴퓨터공학과
{yjkim,cho}@ssrnet.snu.ac.kr

A Mode for Block Ciphers, with Untraceable Dynamic Keys

Yoonjeong Kim Yookun Cho
Department of Computer Engineering, Seoul National University

요 약

블록 암호알고리즘에 대한 기존의 연산 모드들(ECB 또는 CBC 등)은, 각 블록에 대하여 동일한 키로 암호화를 수행한다. 이것은 침입자가 한번의 암호 요청만을 수행하여 많은 수의 평문/암호문 쌍을 얻을 수 있게 함으로써 차분해독법 등의 공격에는 안전성을 제공하지 않는다. 본 논문에서는 블록 암호알고리즘을 위한 새로운 모드를 제안하는데, 이 모드에서는 암호화되는 각각의 블록이 서로 다른 키로 암호화되도록 함으로써 블록의 갯수가 많아짐에 따라 안전성 면에서 상당한 이득을 얻게 된다. 각 블록을 위한 서로 다른 키를 생성하는 것이 추가 연산을 필요로 하지만, 제안하는 모드를 DES에 적용한 TDK(a mode for DES with unTraceable Dynamic Keys)의 수행 시간을 pentium과 sun sparc 상에서 측정해 본 결과 ECB 모드와 거의 유사함을 알 수 있었다.

1. 서론

보안의 중요성이 증대되면서 네트워크를 통한 자료 전송에 암호화 기법이 빈번히 이용되고 있다. 암호화를 통한 통신의 주된 개념은, 키와 같이 높은 안전도를 필요로 하는 자료는 수행시간은 길어도 안전도가 높은 공개키 시스템을 사용하여 전송하고, 일반 자료는 공개키 기법을 통해 전송된 키를 이용하여 비밀키 시스템을 적용하는 것이다. 비밀키 시스템(Secret Key Cryptosystem)은 다시 블록 암호알고리즘과 스트림 암호알고리즘으로 나눌 수 있는데 [1, 2], Data Encryption Standard(DES)는 대표적인 블록 암호알고리즘이다 [1, 3].

DES는 평문/암호문 연관성, 키/암호문 연관성, 암호문의 임의성 등의 특성을 가지며, 키 전수 탐색(exhaustive key search)과 차분 해독법 (differential cryptanalysis) [4, 5, 6], 선형 해독법(linear cryptanalysis) [6, 7, 8] 외의 침입에는 강하다고 알려져 있다. 안전성을 증가 시키기 위한 일환으로 차분해독법 등에 강한 성질을 갖는 새로운 암호 알고리즘들이 제안되거나 [9, 10, 11], 여러 가지 연산 모드 [12, 13, 14]가 지원되고 있다. 그런데, 새로운 암호 알고리즘들은 복적으로 하고 있는 침입 방법에는 강하지만 그 외의 침입에 대한 안전성은 추가의 검증을 필요로 한다. 또한 연산 모드 중 CBC 모드 등은 일부 침입 방법에는 안전하지만 키 전수 탐색 방법이나 차분해독법에 대하여는 안전하지 않으며[2], 최근에 제안된 all-or-nothing encryption 모드 등은 이들 침입에는 안전하지만, 수행 시간이 일반 연산 모드보다 3 배나 큰 단점이 있다 [14].

본 논문에서는 블록 암호 알고리즘을 위한 새로운 모드를 제안하는데, 이 모드는 특히 키 전수 탐색 방법이나 차분해독법에 대하여 일반 연산 모드보다 더 나은 안전성을 지원하면서도 수행 시간은 일반 연산 모드와 유사한 특성을 갖는다. 본 논문의 나머지 부분에서는 이 모드를 DES에 적용한 TDK (a mode for DES with unTraceable Dynamic Keys)에 대한 세부 내용과 안전성 및 수행 성능에 대하여 기술하는데, 비록 이곳에서의 기술은 DES에 국한되지만 제안하는 모드는 기타 다른 블록 암호 알고리즘에도 쉽게 적용할 수 있다.

2. 이전 연구들

2.1 DES에 대한 침입 방법들

블록 암호 알고리즘에 대한 여러 가지 침입 방법이 연구되어 왔는데, 키 전수 탐색은 일반 문자열을 암호화한 암호문 블록 하나가 주어졌을 때 2^{56} 개의 가능한 모든 키에 대하여 의미있는 문자열이 나올 때까지 복호 작업을 수행한다 차분 해독법은 선택된 평문에 의한 공격 (chosen plaintext attack)으로 2^{47}개의 평문/암호문 쌍을 필요로 하는데 [4, 5, 6], 더 많은 수의 평문/암호문 쌍을 이용하면 알려진 평문에 의한 공격(known ciphertext attack)도 수행할 수 있다. 선형 해독법은 키에 대한 선형 근사식을 구성하여 키를 찾는 것으로, 2^{43} 개의 임의 평문에 대한 평문/암호문 쌍을 필요로 한다 [6, 7, 8].

2.2 DES의 연산 모드들

Electronic Code Book (ECB) 모드는 일단의 메시지를 64 비트 블록으로 나눈 후, 각각의 블록을 독립적으로 암호화한다. Cipher Block Chaining (CBC) 모드는 그림 1의 왼쪽 부분과 같이, 초기 벡터 IV와 키 K를 이용하여 n 개의 메시지 블록 $P_1, P_2, P_3, \ldots, P_n$을 암호화하여 암호문 $C_1, C_2, C_3, \ldots, C_n$을 생성한다. 이 때, C_i는 P_i에 C_{i-1}을 exclusive-or한 값을 암호화한, $C_i = DES_K(P_i \oplus C_{i-1})$이다 [12, 13]. 결과적으로 CBC 모드에서는 각각의 암호문 블록이 이전 블록들의 영향을 받아, 동일한 평문도 다른 암호문을 갖게된다. 그러나, 이전 블록들의 영향을 전달하는 매개체가 C_{i-1}로 누구나 알고 있는 값이다. 즉, $P_i = C_{i-1} \oplus DES_k^{-1}(C_i)$로 각 블록에 대하여 키 K에 대한 DES의 입력과 출력인 $P_i \oplus C_{i-1}$과 C_i를 알 수 있으므로, CBC 모드에 의해 암호화된 n 개의 블록에 대하여 n 개의 평문/암호문 쌍이 얻어진다. 이것은 네트워크를 오가는 데이터가 CBC 모드로 암호화된 경우 침입자가 데이타 크기에 비례하는 평문/암호문 쌍을 얻을 수 있음을 의미한다. 즉, 차분 해독법이나 키의 전수 탐색에 대한 CBC 모드의 안전성은 ECB 모드와 동일할 뿐이다 [2].

All-or-nothing encryption 모드는 암호화/복호화되는 n 개의 블록이 모두 연관성을 갖고 있어서 임의의 블록 하나를 구하기 위해 모든 블록을 암호화/복호화해야 하는 특성을 갖는다[14]. 이 모드는 원래 키 전수 탐색에 강하도록 개발된 것인데, 각 블록의

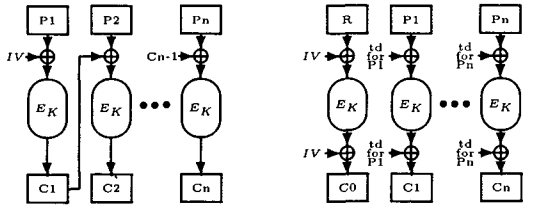

그림 1: CBC 모드와 제안하는 모드

DES 입력이 다른 블록의 영향을 받아 변형됨으로써 차분 해독법에 대하여도 강한 특성을 갖는다. 하지만, 실제 암호화 동작을 제외한 전처리 단계의 수행시간만도 기본 암호 동작의 2 배로 수행 오버헤드가 크다는 문제를 갖고 있다.

3. TDK

TDK의 n 개의 64 비트 평문 블록 $P_1, P_2, ..., P_n$이 주어졌을 때, 우선 임의의 64 비트 난수 블록 R을 생성한다. 그리고, 이 R을 평문 블록들의 맨 앞에 넣은 $n+1$ 개의 블록 $R, P_1, P_2, ..., P_n$에 대하여 암호화를 수행하여 $n+1$ 개의 암호문 블록 $C_0, C_1, C_2, ..., C_n$을 생성한다. 여기서, C_0는 R에 대응하는 암호블록이며, $C_i (i = 1, 2, ..., n)$는 $P_i (i = 1, 2, ..., n)$에 대응하는 암호블록이다.

각 블록의 암호화는 DES와 같이 고정된 키에 의해 수행되며 암호화 과정 앞과 뒤에서 블록마다 값이 다른 동적 키 td의 영향을 받는다. 동적 키 td는 이전 블록의 암호화시에 얻어지는데, DES의 동작에는 전혀 영향을 미치지 않고 td 구성에 필요한 일부 정보만을 얻어올 뿐이다. 이 관계가 그림 1의 오른쪽 부분에 나타나 있다.

우선, 이후의 TDK 세부 내용 기술에 필요한 용어를 정의하면 다음과 같다.

$\oplus$	$\oplus$는 exclusive-or를 의미한다.
$\|\|$	$\|\|$는 접합(concatenation)을 의미한다.
L_i	TDK는 각 64 비트 블록에 대하여 DES를 수행하는데, $L_i (i = 1, .., 16)$는 DES 작업의 i번째 단계 왼쪽 32 비트를 의미한다.
R_i	$R_i (i = 1, ..., 16)$는 i번째 단계의 오른쪽 32 비트를 의미한다.
$\oplus L[j_1, j_2, ..., j_n]$	$\oplus L[j_1, j_2, ..., j_n]$는 $L_{j_1} \oplus L_{j_2} \cdots \oplus L_{j_n}$을 의미한다.
l_i	$l_i (i = 1, ..., 16)$는 복호화시의 i번째 왼쪽 32 비트를 말한다.

이제, td 생성 방법을 기술하면 다음과 같다.

정의 1. 추적 불가능한 동적 키 td는 64 비트 길이로 16 개의 왼쪽 32 비트로부터 다음식에 의하여 얻어진다.

$$td = \oplus L[1, 4, 5, 8, 9, 12, 13, 16] || \oplus L[2, 3, 6, 7, 10, 11, 14, 15].$$

3.1 암호화

각 블록의 암호화 과정은 다음 2 가지만 제외하고는 DES와 동일하다. (1) IP 이전과 FP 이후에, 해당 블록 64 비트를 이전 블록의 암호화시에 생성된 동적 키 td와 exclusive-or 한다. (2) 16 번의 반복 동안 정의 1에 따라 $L_i (i = 1, 2, ..., 16)$ 로부터, 다음 블록을 위한 동적 키 td를 생성한다.

첫 번째 블록을 위한 td는 초기벡터 IV로 초기화되는데, IV는 공개될 수도 있고 공개되지 않을 수도 있다.

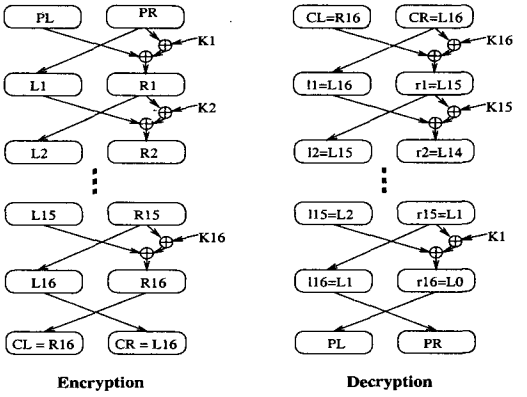

그림 2: 암호화시와 복호화시의 왼쪽 블록들: 암호화시의 $L_1, L_2, ..., L_{16}$은 복호화시의 $l_{16}, l_{15}, ..., l_1 = L_1, L_2, ..., L_{16}$과 동일하다

3.2 복호화

복호화시에 TDK는 우선 $n+1$ 개의 암호문 블록에서 $n+1$개의 평문 블록을 구하는데, 이들 중 첫 번째 평문 블록을 제외한 n 개의 블록들이 원래의 평문 블록이다. 각 블록에 대한 복호화 동작은, DES와 마찬가지로 키 순서가 반대로 되는 것 외에는 암호화 과정과 동일하다. 동적 키 td에 관련하여는 암호화 과정과 복호화 과정이 다를 필요가 없는데 이것은 복호화시에 얻어진 td 값이 암호화시에 얻어진 td 값과 같기 때문이다. 즉, 그림 2에 보이듯 $l_1 = L_{16}, l_2 = L_{15}, ... l_{16} = L_1$이므로, 복호화시에 구한 $\oplus l[1, 4, 5, 8, 9, 12, 13, 16] || \oplus l[2, 3, 6, 7, 10, 11, 14, 15]$이 암호화에서 구한 값 $\oplus L[1, 4, 5, 8, 9, 12, 13, 16] || \oplus L[2, 3, 6, 7, 10, 11, 14, 15]$과 동일하기 때문이다.

TDK의 암호화 과정에서 암호블록 C는 평문블록 P로부터 DES의 키 K와 동적 키 td를 이용하여 $C = DES_K(P \oplus td) \oplus td$를 얻을 수 있고 이를 다시 정리하면 $P = DES_K^{-1}(C \oplus td) \oplus td$를 얻는다. 여기서, DES_K는 키 K를 이용한 DES의 암호화 작업을 말하고 DES_K^{-1}는 키 K를이용한 DES의 복호화 작업을 말한다. 위의 두 식으로부터 암호화과정에서 평문 P로부터 암호문 C가 얻어진 것 같이, 복호화 과정에서 암호문 C로부터 평문 P가 얻어짐을 알 수 있다.

4. 안전성 분석

4.1 td 자체의 안전성

첫 번째 블록을 위한 동적 키 td는 임의의 난수가 키에 의해 변환된 값이고, 마찬가지로 이 이후 블록에 대한 td 값도 난수의 영향을 여전히 받는다. 즉, 동적 키 td는 난수에 기반한 임의 값이므로 td 값으로부터 원래의 키에 대한 정보를 얻는 것은 불가능하다. 더구나, td 값은 연산의 내부에서 사용되는 것으로 침입자에게는 알려지지 않는 값이다.

침입자는 암호문을 DES의 입력으로 하고, 평문을 DES의 출력으로 하여 키를 찾아내려는 시도를 할 수도 있다. 이 경우에는 첫 번째 블록이 임의의 난수가 아니라 알려진 값인 $C0$가 된다. 그러나, $C0$는 임의의 난수에 대응하는 암호문이므로 침입자가 자신이 원하는 특정 값으로 지정하기는 어렵다. 또한, 이 경우에도 td 값은 여전히 침입자에게 알려지지 않는 값이다.

4.2 알려진 침입 방법들에 대한 저항성

DES는 차분 해독법, 선형 해독법, 키 전수 탐색 이외의 침입 방법에는 안전하다고 알려져 있으므로, 여기에서는 제안하는 모드의 이들 침입 방법에 대한 안전성을 기술한다.

16 단계 DES에 대한 차분 해독법은 2^{47} 개의 선택된 평문을 필요로 한다 [4, 5, 6]. TDK 모드가 이용되는 시스템을 침입자가 이용한다고 할 때, 침입자는 극단적으로 한 블록의 암호화를 2^{47}번 수행할 수도 있고 반대로 2^{47} 개의 블록을 한 번에 암호화하

표 1: TDK의 수행 오버헤드

실험환경	ECB (Mbps)	TDK (Mbps)	오버헤드 (%)
pentium	17.1	16.4	4.27
sun sparc	19.6	18.6	5.38

여 대응 암호문들을 구할 수도 있다. 전자의 경우에는, 사용자가 지정한 DES의 입력이 난수에 의해 생성된 td에 의하여 변경된다. 이 난수에 의한 영향을 없애려면, 2^{47} 개의 모든 블록에 대하여 발생한 난수가 동일해야 한다. 64 비트의 난수가 2^{47} 번 같을 확률은 $(1/2^{64})^{2^{47}-1}$ 이며, 따라서 침입자는 차분 해독법을 수행하기 위하여 평균적으로 $2^{64\times(2^{47}-1)}$ 번의 암호화 작업을 다루어야 한다. 후자의 경우에는, 각 블록마다 다른 td 값이 사용자가 지정한 DES의 입력을 변경시켜서, DES의 평문/암호문 쌍에 대한 정보를 얻을 수 없게 한다.

16 단계 DES에 대한 선형 해독법은 2^{43} 개의 알려진 평문/암호문 쌍을 필요로 한다 [6, 7, 8]. 이 경우도 차분 해독법의 경우와 마찬가지로 TDK에서는 침입자에게 보여진 평문/암호문이 난수에 의해 변경되어 DES의 입력/출력이 되므로 침입을 하기 위해 필요한 평문/암호문 쌍을 제대로 얻을 수 없게 된다.

침입자가 i번째 암호문 블록에 대하여 키 전수 탐색을 수행하려면, i 번째 블록에 대한 td 값이 필요하고 이를 계산하기 위해 이전 $i-1$ 개의 블록들도 모두 복호화해야 한다. 첫번째 암호문 블록에 대한 평문 블록은 난수이므로 키 전수 탐색이 의미를 갖지 못한다. 따라서 적어도 두번째 암호문 블록에 대하여 키 전수 탐색을 수행해야 하고, 이것은 한번의 키 검증에 DES 기본 연산을 2 번 필요로 한다. 즉, TDK의 키 전수 탐색 복잡도는 DES보다 적어도 2 배이상이다.

4.3 에러 전파 성질

TDK에서는 암호문 블록 하나가 손상되면 이 이후에 있는 블록들은 복호화를 수행할 수 없다. 이 에러 전파 성질은 좋지 않은 특성이긴 하나, TDK의 암호문을 신뢰성이 있는 전송 수단이나 저장장치를 통해 전송 또는 저장한다면 그리 큰 문제는 되지 않는다.

5. 성능 분석

TDK는 일반 DES보다 다음 3 가지 작업을 추가로 필요로 한다. (1) 난수 생성, (2) td 값 생성을 위한 16 번의 32 bit exclusive-or 연산, (3) td 적용을 위한 2번의 64 bit exclusive-or 연산이다. 표 1은 DES ECB 모드에 대한 TDK의 수행 오버헤드를 보여준다. 실험은 pentium 환경 (64 Mbyte 메모리가 있는 pentium 166 MHz 상의 linux Kernel 2.0에서 동작하는 gcc version 2.7.2를 이용)과 sun sparc 환경 (64 Mbyte 메모리를 가진 sun sparc 167 MHz 상의 solaris 5.5에서 동작하는 gcc version 2.8.1을 이용)에서 수행하였다. DES 코드는 Eric Young의 libdes이며 [16], TDK 코드는 여기에 연산들을 추가하여 구성하였다. 표 1에서 ECB 열과 TDK 열은 각각의 수행 속도(Megabits/second)를 의미하며, 오버헤드 열은 TDK의 ECB 모드에 대한 수행 시간 오버헤드로 $\frac{encryption\ time\ of\ TDK - encryption\ time\ of\ ECB}{encryption\ time\ of\ ECB} \times 100$ 에 의해 구한 값이다. Pentium환경과 sun sparc 환경에서 모두 TDK의 수행 오버헤드가 5 % 내외임을 알 수 있다.

6. 결론

본 논문에서는 블록 암호 알고리즘을 위한, 추적 불가능한 동적 키를 갖는 새로운 연산 모드를 제안하였고 이를 DES에 적용한 TDK에 대한 안전성 및 성능 분석을 기술하였다. TDK는 침입자가 알 수 있는 암호문을 이용하는 CBC 모드와 달리 난수에 기반하여 값을 알 수 없는 동적 키 td를 이용하여 각 평문 블록을 각기 다른 키로 암호화한다. TDK는 수행 시간은 DES의 ECB 모드와 거의 유사하면서도 DES가 취약한 차분해독법, 선형해독법 및 키 전수 탐색 등에서는 기존 모드들보다 안전한 특성을 갖는다.

References

[1] Douglas R. Stinson, *CRYPTOGRAPHY Theory and Practice*, Boca Rayton, CRC Press, Inc., 1995.

[2] RSA Laboratories, *Cryptography FAQ (Frequently Asked Questions)*, http://www.rsasecurity.com/rsalabs/faq, 1998.

[3] National Bureau of Standards, *Data Encryption Standard (DES)*, Fedral Information Processing Standards Publication 46-2, December 30, 1993.

[4] Eli Biham, Adi Shamir, "Differential Cryptanalysis of DES-like cryptosystems", *Advances in Cryptology-CRYPTO'90*, Lecture Notes on Computer Science, **537**, pp.2–21, 1991.

[5] Eli Biham, Adi Shamir, "Differential Cryptanalysis of the Full 16-round DES", *Advances in Cryptology-CRYPTO'92*, Lecture Notes on Computer Science, **740**, pp. 487–496, 1993.

[6] Ronald L.Rivest, *Computer and Network Security Lecture Notes*, Part 4. Block and Stream Ciphers, http://web.mit.edu/6.857/www/home.html, 1998.

[7] Mitsuru Matsui, "Linear Cryptanalysis method of DES cipher", *Advances in Cryptology-EUROCRYPT'93*, Lecture Notes on Computer Science, **765**, pp.386–397, 1994.

[8] Mitsuru Matsui, "The first experimental cryptanalysis of the Data Encryption Standard", *Advances in Cryptology-CRYPTO'94*, Lecture Notes on Computer Science, **839**, pp. 1–11, 1994.

[9] Carlisle M. Admas, "On immunity against Biham and Shamir's "differential cryptanalysis"", *Information Processing Letters*, vol. 41, pp. 77–80, 1992.

[10] Kaisa Nyberg, Lars Ramkilde Knudsen, "Provable Security against differential cryptanalysis", *Advances in Cryptology-CRYPTO'92*, Lecture Notes on Computer Science", **740**, pp. 566–574, 1993.

[11] Toshinobu Kaneko, Kenji Koyama, Routo Terada, "Dynamic Swapping Schemes and Differential Cryptanalysis", *IEICE Transactions on Fundamentals*, vol. E77-A, no. 8, pp. 1328–1335, August, 1994.

[12] Gilles Brassard, *Modern Cryptology - A tutorial*, Lecture Notes on Computer Science, vol. 325, 1988.

[13] National Bureau of Standards, *DES Modes of Operations*, Fedral Information Processing Standards Publication 81, December 1980.

[14] Ronald L. Rivest, "All-Or-Nothing Encryption and The Package Transform", *Fast Software Encryption*, Lecture Notes on Computer Science, **1267**, 1997.

[15] Eli Biham, "On modes of Operation", *Fast Software Encryption*, Lecture Notes on Computer Science, vol. 809, pp. 116-120, 1994.

[16] Eric Young, libdes, ftp://ftp.psy.uq.oz.au/pub/Crypto/DES, 1997.