개인 클라우드 컴퓨팅 환경을 위한 스프레드 랜덤 인터리버 기반의 효율적인 DES 알고리즘

정연호*

A Spread Random Interleaver based Efficient DES Algorithm for Personal Cloud
Computing Environments

Yeon Ho Chung*

이 논문은 2011학년도 부경대학교의 지원을 받아 수행된 연구임 (PK-2011-2-34)

요 약

본 논문에서는 개인기반 클라우드 컴퓨팅 환경에서 복잡하고 시간이 많이 소요되는 3DES 알고리즘 리즘에 비해 기존의 DES 암호화 알고리즘을 기반으로 안정성과 속도를 개선한 알고리즘을 제안한다. 제안하는 알고리즘은 DES 알고리즘에 있어 안전성 측면에서 단점인 초기치환 및 최종치환을 위한 고정된 치환테이블을 스프레드 랜덤 인터리버 기법을 적용한 동적 치환박스를 적용하여 보안성을 제고하였다. 기존 알고리즘과의 시뮬레이션 결과 비교를 통해 제안하는 알고리즘을 활용할 경우 최소한의 변경으로 전달 데이터에 대한 기밀성과 안정성을 확보하고 빠른 속도로 암호화 및 복호화를 달성할 수 있음을 확인할 수 있었으며 CBC 모드를 적용하여 가변 길이의 데이터에 대해서도 효과적인 암호화가 가능하다.

ABSTRACT

An efficient encryption algorithm based on the Data Encryption Standard (DES) for personal cloud computing environments is presented. The proposed algorithm improves data privacy, security and also encryption speed, compared with the triple DES. The improvement of the proposed algorithm stems from enhanced privacy inherent from the use of spread random interleaver in the place of the known substitution table for initial and final permutations in the DES algorithm. The simulation results demonstrate that the interleaver based DES (I-DES) is found to run faster than the triple DES algorithm and also offer improved security. The proposed algorithm also offers encryption for variable-length data using the Cipher Block Chaining (CBC).

키워드

스프레드 랜덤 인터리버, DES, 3DES, 암호 알고리즘

Key word

Spread random interleaver, DES, 3DES, Encryption algorithm

---

* 정회원 : 부경대학교 정보통신공학과

접수일자 : 2012. 11. 15

심사완료일자 : 2012. 12. 10

I. 서 론

최근 인터넷의 활용의 증가와 응용 프로그램 관련 콘텐츠 및 데이터 대용량화에 따라 정보의 저장과 처리에 있어 응용 서비스와 관계된 인터넷 데이터 센터(IDC : Internet Data Center)의 중요성이 점차 증대되고 있다. 그래서 이들을 활용하는 일반 사용자들은 다양한 컴퓨터 자원을 효율적이고 또한 경제적으로 사용하고자 하는 요구가 점증하고 있다. 이를 충족하기 위해 주목을 받고 있는 기술이 개인기반 클라우드 컴퓨팅(Personal Cloud Computing)이다[1].

현대 암호 알고리즘 중 대표적인 대칭 암호화 알고리즘 중의 하나인 DES 알고리즘의 경우 암호화 키 크기가 56비트(패리티 비트 제외)로 전수 조사를 할 경우 256개 키를 조사함으로써 구할 수 있게 된다. 그러나 현재의 기술로 병렬처리가 가능한 컴퓨터를 이용할 경우 단시간에 모든 경우의 수를 검색할 수 있는 단점을 가진다. 한편, 이를 보완하는 3DES와 AES 알고리즘이 제안되고 있으나 개인기반 클라우드 컴퓨팅 환경의 활용에서는 자체의 복잡성으로 인하여 계산에 있어 다소 시간이 많이 걸리는 단점을 가지고 있다[2,3].

본 논문에서는 개인기반 클라우드 컴퓨팅 환경에서 적용하기 적합한 DES 암호화 알고리즘 기반의 개선된 알고리즘을 제안한다. 제안된 암호기술의 핵심은 DES 알고리즘에 있어 안전성 측면에서 의미를 가지지 못하는 초기치환 및 최종치환을 위한 고정된 치환테이블(substitution table)을 스프레드 랜덤 인터리버(spread random interleaver) 방법을 사용한 동적인 치환박스를 활용하여 보안성을 높이는 개념이다. 따라서, 전달 데이터에 대한 기밀성과 안정성을 확보하고 빠른 속도로 암호화 및 복호화를 달성한다.

그리고 개인기반 클라우드 컴퓨팅 환경에서 다양한 크기의 전달 메시지를 수용하기 위해 블록암호 운영 모드 중 하나인 CBC (Cipher Block Chaining) 블록 암호화 운영 모드를 적용하여 암호화되는 텍스트 길이가 가변적인 경우에도 적용 가능하도록 한다.

본 논문에서 제안하는 동적인 치환박스 생성에 활용된 스프레드 랜덤 인터리버 방법은 무선 환경에서 주로 데이터의 랜덤화에 적용되는 방법으로 잡음에 강하고 높은 기밀성을 제공하는 특징을 가지고 있으며 난수 생

성은 BBS(Blum Blum Shub) [4] 의사난수 생성 방식을 적용한다.

II. DES 알고리즘

DES 알고리즘은 대칭키 기반의 암호화 알고리즘으로 블록 암호화 방식을 사용하여 전송 정보의 해당 64비트 평문 블록에 적용되어 동일한 길이의 암호문을 생성한다. 이때 동일한 길이의 암호화 키가 암호화와 복호화에 사용된다[5].

DES 알고리즘의 암호화 과정은 두 개의 치환(P-박스 : permutation box)과 16개의 Feistel 라운드 함수(round function)로 구성된다[6]. 두 개의 치환박스는 라운드 함수가 시작되기 전과 후에 사용되며 각각 초기치환, 최종치환이라고 한다. 이 방법에서 각 라운드는 암호화 키를 이용하여 키 생성기에 의해 생성된 48비트 길이를 가지는 라운드 키를 사용한다.

그림 1. DES 알고리즘 구조
Fig. 1 DES algorithm structure

그림 1에서는 DES 알고리즘의 구조를 나타내고 있다. 먼저 64비트 평문(64-bit plaintext)이 입력되고 56비

트 크기의 암호화 키(56-bit cipher key)가 사용된다. 여기서 암호화 키는 최초에 64비트가 입력이 되나 패리티 비트(parity bit) 제거과정을 거쳐 실제로는 56비트가 사용된다. 그리고 입력된 암호화 키는 라운드 키 생성기(Round-key generator)를 통하여 각 라운드에 사용될 라운드 키를 생성하며, 입력된 평문은 초기치환(Initial permutation)을 거쳐 라운드 함수를 수행하고 최종치환(Final permutation)을 통하여 암호화된 64비트 암호문(64-bit ciphertext)이 생성된다.

여기서, 기존 DES 알고리즘의 초기치환 및 최종치환은 암호 알고리즘으로써의 의미를 가지지 못하며, 암호화 및 복호화 속도 면에서 20% 정도 지연시키는 단점을 가지고 있다 [7]. 또한 암호화 키의 크기가 56비트로 짧아, 전수조사를 할 경우 현재의 기술로 병렬처리가 가능한 컴퓨터를 이용할 경우 단시간에 모든 경우의 수를 검색할 수 있는 단점을 가진다.

이와같은 단점을 보완하기 위해 3DES와 AES 알고리즘이 개발되었으나, 3DES 알고리즘은 DES 알고리즘을 세 번 수행하게 되므로 물리적으로 3배 정도의 시간이 소요되게 되며, AES 알고리즘은 수학적 기반에 의하여 구현되므로 DES 알고리즘 보다 연산속도가 느리다는 단점이 있다.

그러므로 DES 알고리즘에서 안전성 측면에서 의미를 가지지 못하는 초기치환 및 최종치환을 위한 고정된 치환테이블을 대신하여 랜덤성 강화를 통한 보안성을 높일 수 있는 스프레드 랜덤 인터리버를 사용한 동적인 치환박스를 사용한다.

III. 스프레드 랜덤 인터리버 기반의 DES 알고리즘

3.1. 스프레드 랜덤 인터리버

인터리버는 전송되는 데이터를 시간이나 주파수 영역에서 데이터를 분산시키거나 데이터의 위치를 섞어 줌으로써, 복조 이후에 발생하는 연접오류(burst error)를 산발오류(random error)로 바꾸어주어 오류정정을 용이하게 하는 역할을 한다 [8].

본 논문에서는 보안성을 높이기 위해 높은 무작위성을 가지는 스프레드 랜덤 인터리버를 적용하였으며 이는 연속하는 데이터에 대해 인터리빙을 수행할 때 각각

의 비트들의 위치가 S만큼 떨어져 위치하도록 인터리버를 구성함으로써 인터리빙 성능을 극대화하는 방법이다[9].

스프레드 랜덤 인터리버의 알고리즘에서 사용되는 위치 값들은 BBS 의사난수 생성기를 통하여 생성된 의사난수를 이용한다.

스프레드 랜덤 인터리버의 알고리즘의 동작과정은 다음과 같다. 먼저 위치 값을 생성(generates position values)하여 S개 이전동안 생성된 비트들의 위치 값들과 비교(checks condition)하여 거리가 $\pm S$ 안에 있으면 현재의 위치 값을 버리고 다시 위치 값을 생성하고, 거리가 $\pm S$ 이상 떨어져 있다면 위치 값을 저장(saves position values)한다. 위의 과정을 데이터의 비트수인 N개가 저장될 때까지 반복한다. 위에서 설명한 과정을 통해서 생성된 N개의 위치 값들을 이용하여 비트들을 재배열(rearrange bits)하여 인터리빙 과정을 완료한다.

예를 들어, S가 5이고 첫 번째 생성된 위치 값이 20이라면, 두 번째 생성된 위치 값이 15에서 25 사이의 값일 경우 새로운 위치 값을 생성하고 그렇지 않을 경우 저장하고 다음에 저장할 위치 값을 생성한다.

모든 N개의 위치 값을 얻는데 걸리는 시간은 위치 값들의 간격 S가 증가함에 따라 증가하게 되므로, 일반적으로 식 (1)을 만족하도록 S의 크기를 결정하게 된다 [10]. 이 과정을 그림 2에서 보여주고 있다.

$$S < \sqrt{N/2} \quad (1)$$

그림 2. 스프레드 랜덤 인터리버
Fig. 2 Spread random interleaver

3.2. BBS 의사난수 생성기

본 연구에서 사용된 스프레드 랜덤 인터리버의 위치 값 생성을 위하여 의사난수 생성 알고리즘 중의 하나인 BBS 의사난수 생성기를 사용하였다. 구체적인 난수 생성과정은 다음과 같다.

먼저 $4k + 3$ 형태(k 는 임의의 정수)를 갖는 두 개의 매우 큰 소수 p 와 q 를 구하고 그림 3의 n 값은 p 와 q 의 곱으로 구한다. 그 다음, n 과 서로 소(relative prime)인 임의의 수 r 를 선택하게 된다.

최초의 종자 값(seed)으로 쓰일 x_0 의 값은 r^2 를 n 으로 나눈 나머지($r^2 \bmod n$)를 계산하여 구한다. 그 다음의 종자 값(x_{i+1})은 현재의 종자 값(x_i)의 제곱을 n 으로 나눈 나머지($x_i^2 \bmod n$)를 이용하여 구한다. 위와 같은 과정을 통하여 생성된 난수 정수열의 최하위 비트(Least Significant Bit)를 추출하여 그것을 의사난수 비트로 사용하게 된다.

여기서, p 와 q 값이 노출되어 있다면 가능한 x_0 값을 이용하여 i 번째 비트 값을 알아낼 수 있다. 그러나 n 이 큰 수라면 이 수열은 예측 불가능하여 안전한 수열이 되며 n 의 값이 크다면 이전의 모든 비트를 알고 있다고 하더라도 다음 비트 값을 추출할 수 없으며 각 비트가 0이 되거나 1이 될 확률은 약 50%로 우수한 성능을 가지게 된다.

그림 3. BBS 의사난수 생성 과정
Fig. 3 BBS pseudo-random number generator

3.3. 스프레드 랜덤 인터리버 기반의 DES 알고리즘

기존의 DES 알고리즘은 치환테이블이 고정되어 있어 같은 평문에서 초기치환 과정을 수행할 경우 항상 같은 치환 결과가 도출되어 DES 알고리즘의 안정성 측면에서 단점으로 여겨지고 있다. 본 연구에서는 전술한 의사난수 생성기를 이용하여 초기치환과 최종치환 테이블을 생성하므로 같은 평문을 치환할 경우 264가지의 경우의 수가 발생되고 이를 16개의 라운드 함수를 거쳐 암호화가 진행이 된다.

그림 4. I-DES 알고리즘의 암호화 및 복호화 과정
Fig. 4 Encryption and decryption procedure of I-DES algorithm

그림 4에서 제안한 스프레드 랜덤 인터리버 기반의 DES 알고리즘(I-DES)의 암호화 및 복호화 과정을 보여주고 있다. 그림에서 보면 암호화가 시작되면 각 라운드에 사용될 라운드 키들을 암호화 키를 이용하여 생성(generate round-keys)한다. 이후, I-DES 알고리즘에서는 BBS 의사난수 생성기에 의해 생성된 난수를 이용한 스프레드 랜덤 인터리버를 이용하여 치환테이블을 생성한다. 생성된 치환테이블에 따라 64비트 길이의 평문(64-bit plaintext)을 초기치환(Initial permutation) 과정을 수행하고, 이후 16번의 라운드(Round)를 수행한다. 이를 다시 원래의 배열 순서로 치환하기 위하여 최종치환(Final permutation) 과정을 수행하게 되면, 최종적으로 암호화된 64비트 길이의 암호문(64-bit ciphertext)이 완

성된다.

복호화 과정은 암호화 과정과 유사하나 각 라운드에서 사용되는 라운드 키가 역순으로 사용된다.

대칭키 암호화 기법은 일반적으로 블록 암호를 기반으로 구현이 이루어진다. 따라서 DES 알고리즘도 고정된 길이의 데이터만을 암호화와 복호화 할 수 있으며 64 비트 길이의 블록을 암호화 하거나 복호화 할 수 있다 [12]. 이러한 고정길이의 데이터 암호화 및 복호화 문제점을 해결하기 위해 블록 암호화 운용방식의 하나인 CBC 알고리즘을 이용하여 가변길이를 가지는 데이터에 대해서도 암호화가 가능하도록 하였다. 그림 5는 CBC 알고리즘의 순서도이다.

그림 5. CBC 알고리즘의 블록 다이어그램
Fig. 5 Block diagram of CBC algorithm

위의 다이어그램에서 사용된 P, IV, K, C 의 정의는 식 (2)과 같다.

$$\begin{aligned} P &= \{P_1, \dots, P_i\} \\ K &= \{K\} \\ C &= \{C_1, \dots, C_i\} \\ IV &= \{IV\} \end{aligned} \quad (2)$$

P 는 암호화된 평문 블록이며 8바이트 단위로 블록화되어있다. K 는 송신자 및 수신자가 알고 있는 암호화 키이며, C 는 최종 암호화된 블록이고 각 블록에서 암호화 과정을 거쳐 생성된 암호블록들의 합이다. IV (Initial Vector)는 초기화 벡터이며, E 는 암호화를 의미한다.

CBC 알고리즘은 각각의 블록을 암호화 하기 위해 이전블록의 암호화된 블록을 이용한다. 그러나 최초의 평문 블록은 이전블록의 암호화된 블록이 존재하지 않기 때문에 암호화를 위해 IV 라는 초기값을 사용하게

된다. 본 연구에서 사용된 IV 는 일반적인 정의에서 사용되는 '0' 값을 가지는 8바이트 길이의 고정 IV 를 사용하였다.

CBC 알고리즘에서는 입력되는 평문의 길이가 블록 길이인 64의 배수가 아닐 경우에는 영 채우기(zero-padding)를 통하여 64의 배수로 만든 후 DES 알고리즘의 암호화 과정을 수행하도록 하였다. 이 과정을 그림 6에서 보여주고 있다.

그림 6. CBC 알고리즘
Fig. 6 CBC algorithm

IV. 성능 분석

4.1. 테스트 베드 구현

제안한 I-DES 알고리즘을 구현하고 CBC 알고리즘을 적용하여 데이터가 정상적으로 전송됨을 확인하기 위해서 인터넷 환경에서 시뮬레이션 시스템을 구축하였다.

데이터 전송을 위하여 TCP 프로토콜을 사용하였고, 바이트 형(byte type)의 데이터를 송수신하였다. 시스템은 서버와 클라이언트로 나누어지며 동작 과정은 그림 7에서 보여주고 있다.

서버 측에서는 소켓을 생성(Create socket)하고 IP와 포트번호를 바인딩(binding) 하는 과정을 수행하여 클라이언트를 기다리게(Listen) 된다. 클라이언트가 접속(Connect)을 하게 되면 이를 받아서(Accept) 연결이 되고 데이터를 송수신(Receive & Send)할 수 있게 된다. 데이터 송신 시에는 평문을 암호화(Encryption)하여 전송하고 수신 시에는 받은 데이터를 복호화(Decryption)를 진

행하며 이를 반복한다.

수신측에서는 소켓을 생성하고 서버의 IP를 이용하여 접속하게 된다. 서버와 마찬가지로 송신 시에는 보낼 평문을 암호화 과정을 거쳐 전송하며 수신된 암호문을 복호화 과정을 수행한다.

그림 7. 테스트베드 시스템 동작 과정
Fig. 7 Operating procedure of the test-bed system

구현한 시스템은 서버와 클라이언트로 기능이 구분되며 서버로 시작 시 클라이언트가 접속할 때까지 대기하게 된다. 클라이언트로 시작 시 서버의 IP 주소를 이용하여 접속할 수 있도록 하였다. 암호화 키는 8바이트 길이로 설정하며, 서버와 클라이언트가 사전에 정해진 값을 사용하게 된다.

서버와 클라이언트는 각각 평문을 입력받아 입력한 암호화 키를 이용하여 평문을 암호화하여 상대방에게 전송하고, 상대방은 받은 암호문을 동일한 키를 이용하여 복호화 과정을 수행한다.

그림 8에서는 정상적으로 송수신이 이루어진 경우와 키 값이 일치하지 않는 경우를 보이고 있다. 수신 시에는 암호문을 수신하게 되고 이를 평문으로 복호화 과정을 수행한다. 이때, 잘못된 암호화 키를 사용할 경우 복호화가 정상적으로 이루어지지 않음을 확인할 수 있다. 송신 시에는 입력받은 평문을 암호화 키를 이용하여 암호화 과정을 수행한 암호문을 상대방에게 전송하게 된다.

그림 8. 암호문의 송수신
Fig. 8 Transmission and reception of ciphertext

4.2. 실험결과 및 고찰

구현한 테스트베드를 이용하여 제안한 I-DES 알고리즘의 암호화 및 복호화 속도에 관하여 성능을 분석하였다. 먼저 기존의 고정된 치환테이블을 이용하는 방법과 제안한 스프레드 랜덤 인터리버를 이용하여 생성한 치환테이블을 생성하는 방법을 사용하였을 때의 치환속도를 비교하였다. 200회에서 4,000회까지 치환과정을 수행하였다. 그림 9에서와 같이 2,000회까지는 스프레드 랜덤 인터리버를 이용한 방법이 보다 많은 시간이 걸리나 그 이후로는 인터리빙 시간이 치환수행에 있어 최초에 한번만 수행하므로 영향이 크게 없어 유사한 수행속도를 보이고 있다.

그림 9. 기존 치환테이블과 스프레드 랜덤 인터리버를 사용한 치환테이블간의 치환속도 비교
Fig. 9 Comparison of existing permutation table and spread random interleaver based permutation table with permutation speed

그림 10 및 그림 11에서는 CBC 알고리즘을 적용한 DES 알고리즘과 I-DES 알고리즘의 암호화 및 복호화 수행시간을 측정한 결과를 보여주고 있다. 그림에서 보면 기존의 DES와 I-DES 알고리즘의 비교에서 큰 차이가 없음을 보이고 있다.

또한 CBC 알고리즘을 적용하였을 경우에 보다 효율적인 암호화 및 복호화가 이루어지기 때문에 적용하지 않았을 때보다 효율이 높음을 확인할 수가 있었다.

그림 10. CBC 알고리즘을 적용한 I-DES와 DES의 암호화 속도 비교

Fig. 10 Comparison of proposed CBC-based I-DES and DES with encryption speed

그림 11. CBC 알고리즘을 적용한 I-DES와 DES의 복호화 속도 비교

Fig. 11 Comparison of proposed CBC-based I-DES and DES with decryption speed

V. 결 론

기존의 DES 알고리즘은 취약한 보안성 및 구현속도로 인해 개인기반 클라우드 컴퓨팅에 적용하기가 곤란한 문제점을 가지고 있다. 이를 개선한 알고리즘으로는 3DES 알고리즘이 제안되고 있으나 암호화 및 복호화를 3번 수행해야 하므로 많은 시간과 비용이 발생하게 되고, AES 알고리즘의 경우 암호화 과정이 수학적 과정에 기반 하여 수행되어 DES 알고리즘보다 느린 암호화 및 복호화 속도를 가진다. 본 논문에서는 이러한 문제점을 해결하고 개선된 DES 알고리즘으로 보안성이 높은 스프레드 랜덤 인터리버를 적용하여 보안성과 암호화 및 복호화 속도를 높이는 효과를 얻을 수 있는 스프레드 랜덤 인터리버 기반의 DES 알고리즘 (I-DES)를 제안하였다. I-DES 알고리즘과 기존의 DES 알고리즘의 비교분석에서 수행속도에서 장점을 확인할 수 있었고 확산성 면에서도 안전도가 높음을 알 수 있었다. 또한 실제 인터넷 환경에서의 검증을 위하여 제안한 알고리즘을 블록 암호화 운영 방식의 하나인 CBC 알고리즘에 적용하여 정상적으로 동작됨을 확인하였다.

참고문헌

- [1] 김지연 외 3명, “클라우드 컴퓨팅 환경의 가상화 기술 취약점 분석연구”, 한국정보보호학회, 제19권 제 4호, pp. 72-77, 2009
- [2] Behrouz A. Forouzan, “암호학과 네트워크 보안”, Mcgraw-Hill Korea, 2008
- [3] 신승수, 한군희, “암호화된 데이터베이스에서 인덱스 검색 시스템 구현”, 한국산학기술학회논문지, pp.1653-1660, 2010
- [4] L.Blum, M.Blum and M.Shub, “A Simple Secure Pseudo-Random Number Generator,” SIAM Journal of Computing, Vol.15, No.2, pp.364-383, May 1986
- [5] William M. Daley, Raymond G. Kammer, NIST, 1999
- [6] 원동호, “현대 암호학”, 도서출판 그린, pp.3, 2003
- [7] W.Diffie and M.E.Hellman, “Exhaustive Cryptanalysis of the NBS Data Encryption Standard,” IEEE, Vol.10, No.6, pp.74-84, 1977

- [8] Juha Heiskala, “OFDM 무선랜”, 브레인코리아, 2003
- [9] D. Divsalar, F. Pollara, “Multiple Turbo Codes for Deep-Space Communication”, The JPL TDA Progress Report, pp.42-121, 1995
- [10] 진익수, 노예철, 주유상, 강범주, “터보 부호의 인터리버 분석”, 정보통신산업진흥원, 주간기술동향 905호, 1999
- [11] Thomas M. Cover, Joy A. Thomas, “Elements of Information Theory”, Wiley, 2005

저자소개

정연호 (Yeon Ho Chung)

1984년 경북대학교
전자공학과(공학사)
1992년 The Imperial College, Univ.
of London, U.K.(공학석사)

1996년 Liverpool University, U.K. (공학박사)
1995년 영국 Freshfield Comm. Ltd. 기술위원
2004년 영국 Plymouth 대학교 초빙연구원
2006년 미국 Pennsylvania 주립대학교 객원교수
2008년 부경대학교 국제교류원장
2001년-현재 부경대학교 정보통신공학과 교수
※ 관심분야: 무선접속기술, OFDM, 가시광선통신,
적응변조, 부호화기술