DES의 고속화 실현을 위한 치환연산과 대치 연산의 병렬처리 방법

손기욱, 박응기
한국전자통신연구원

The Parallel Processing of Permutation and Substitution for the High-Speed DES

Kiwook Sohn, Eungki Park
Eelectronics and Telecommunications Research Institute

요 약

DES 암호 알고리즘은 정보의 기밀성 서비스와 무결성 서비스 실현을 위해 널리 사용되고 있다. DES를 하드웨어로 실현이 곤란한 분야에서는 소프트웨어로 구현하여 사용되고 있으나 처리 속도의 문제로 인해 사용하지 못하는 경우도 존재한다. 본 논문에서는 소프트웨어의 처리 속도 문제를 해결하기 위해 DES 암호 알고리즘의 치환 연산과 대치 연산을 병렬로 처리하는 방법을 제시하여 고속으로 정보를 실시간으로 보호하고자 하는 분야에 적용할 수 있도록 하였다.

I. 서 론

컴퓨터를 이용한 중요한 정보의 저장,처리,전송 등이 필요함에 따라 그 정보를 안전하게 보호하기 위한 많은 방법들이 연구되고 있다. 그 중 암호화를 이용한 정보의 보호 방법이 일반적으로 많이 사용된다. 그러나 이 방법은 각종 통신 시스템, 컴퓨터 시스템에서 데이터 처리 중 암호화를 부가적으로 수행함에 따라 암호화 과정에서 병목현상 문제가 발생될 수 있다. 고속 컴퓨터 시스템, 고속 통신망 등을 구성하기 위해서는 고속 암호화 처리는 필수적으로 해결되어야 될 과제이다. 현재 고속으로 정보를 보호하는 분야에서는 하드웨어로 구현된 DES(Data Encryption Standard)[1] 칩을 주로 이용하고 있으나[2][3][4] 그 사용이 제한적이며, 하드웨어로 구현이 곤란한 분야는 소프트웨어로 구현[5]하여야 하는데 DES 알고리즘 처리 속도에 문제가 있다[6].

따라서 본 고에서는 널리 알려진 단일 키를 사용하는 블럭 암호화 방식인 DES를 일반화하여 고속으로 처리하기 위한 방안을 제시한다. 본 고의 주요 내용은 소프트웨어로 DES 암호 알고리즘을 처리할 경우 속도의 향상을 피하고자, DES의 암호 알고리즘의 치환 연산과 대치 연산을 병렬로

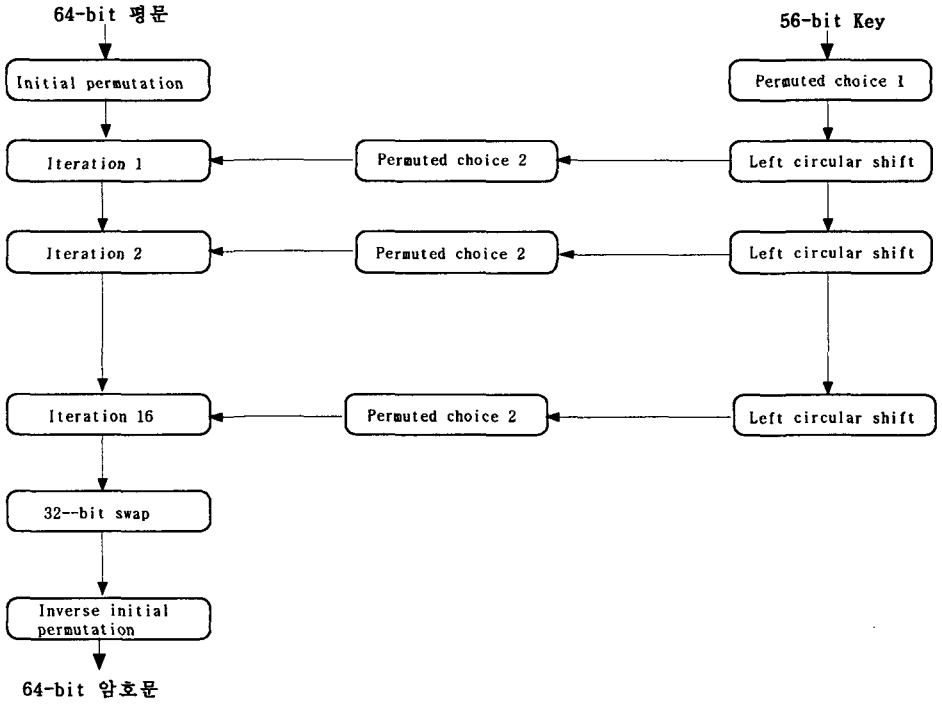
처리하는 방법을 제시하고자 한다.

II. DES 암호 알고리즘[1]

1. DES 기본 원리

DES 암호 알고리즘은 1977년 미국의 NBS(국립표준국;현재의 NIST)에서 미국 표준 암호 알고리즘으로 발표하여 가장 널리 쓰이고 있는 암호화 알고리즘이다. DES는 0,1의 2진 데이터를 암호화하는 블럭 암호화 방식을 채용하고 있다. DES 암호 알고리즘은 데이터를 64비트 블럭마다 각 블럭을 치환(Permutation) 연산과 대치(Substitution) 연산을 반복시켜 암호화 한다. 키는 64 비트지만 그 가운데 8비트는 오류 검출을 위한 검사 비트로 사용되며, 암호화 및 복호화에 사용하는 비트는 56 비트이다. 이 키에 의해 각 단의 XOR(Exclusive OR) 연산이 실행된다.

DES 암호 알고리즘은 우선 64비트를 초기 치환(Initial Permutation) 연산을 취한 후 64비트가 32비트씩 우측과 좌측으로 나누어져 16회에 걸쳐 치환 연산, 대치 연산 및 XOR 연산을 반복한다. 다음 마지막으로 한번 더 역 초기 치환(IP-1 : Inverse Initial Permutation) 연산으로 암호화를 종료한다. 복호화는 암호화와 동일하며, 다만 키의 입력을 역순으로 한다.



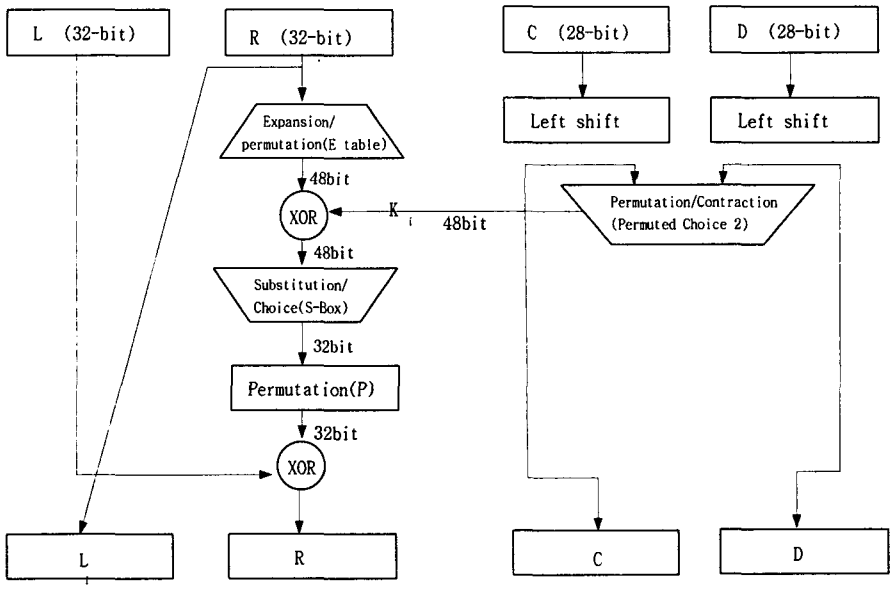
(그림 1) DES의 기본 원리

2. 반복 과정(Iteration)의 설명

$L_i = R_{i-1}$

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

* P : Permutation, S : S-box, E : Expansion permutation



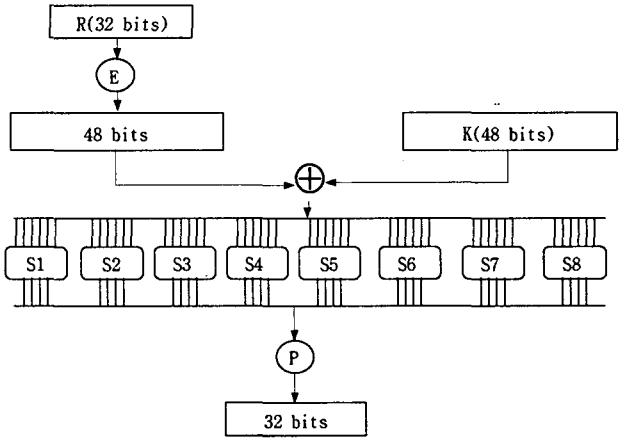
(그림 2) 16회 반복 과정

3. 대치 연산(S-Box)

대치 연산을 수행하는 8개의 S-Box로 구성되어 있으며, 각각 6비트를 입력으로 받아 4비트의 출력 갖는다.

o Function f의 구성, f(R,K)

각 S-Box의 입력 6비트 중 첫번째 비트와 마지막 비트는 2비트의 조합으로 S-Box의 Row를 결정하고, 중간에 있는 4비트는 Column을 결정한다.



(그림 3) S-Box 처리

III. DES의 병렬 처리

DES 암호 알고리즘은 (그림 1), (그림 2) 그리고 (그림 3)에서 보는 바와 같이 주로 치환 연산과 대치 연산으로 구성되어 있다[1]. 본 고에서는 치환 연산과 대치 연산을 병렬로 처리하는 방법을 제시하여 고속으로 정보를 보호하고자 하는 분야에 적용 가능하게 하고자 한다.

1. 치환 연산의 병렬 처리

본 고에서 DES 암호 알고리즘의 병렬 처리의 대상은 초기 치환 연산, 역 초기 치환 연산, 확장 치환 연산이다. 확장 치환 연산은 64 비트의 평문 데이터를 암호화 혹은 복호화하기 위해 각각 16회의 실행을 한다. 이 부분은 DES 암호 알고리즘의 주요 부분 중의 하나로 이러한 치환 연산을 병렬로 처리한다면 소프트웨어로 구현해야 하는 분야에서는 실시간으로 고속의 정보를 보호할 수 있다. 이러한 치환 연산은 입력으로 받은 비트의 위치 값을 항상 정해진 위치의 비트 값과 서로 치환하는 행위를 한다.

가. 초기 치환

초기 치환은 64 비트의 평문 데이터 64 비트를 입력 받아 64 비트의 치환 결과를 출력한다. 초기 치환의 결과는 (그림 4) 와 같이 8 비트의 단위로 8개의 블록이 일정한 형태를 갖고 있다.

따라서 본 고에서는 8개의 블록을 각각 프로세서(혹은 프로세스)에 할당하여 병렬로 처리하여 DES 암호 알고리즘의 처리 속도를 향상시킨다.

치환 결과 값	블 록	할당프로세서(프로세스)
M58 M50 M42 M34 M26 M18 M10 M2	블록 1	P1
M60 M52 M44 M36 M28 M20 M12 M4	블록 2	P2
M62 M54 M46 M38 M30 M22 M14 M6	블록 3	P3
M64 M56 M48 M40 M32 M24 M16 M8	블록 4	P4
M57 M49 M41 M33 M25 M17 M9 M1	블록 5	P5
M59 M51 M43 M35 M27 M19 M11 M3	블록 6	P6
M61 M53 M45 M37 M29 M21 M13 M5	블록 7	P7
M63 M55 M47 M39 M31 M23 M15 M7	블록 8	P8

(그림 4) 초기 치환의 병렬 처리

나. 역 초기 치환

역 초기 치환은 초기 치환을 거쳐 (그림 2)와 같은 연산을 16회의 반복한 결과 값인 64 비트를 입력 받아 64 비트의 치환 결과를 출력한다. 역 초기 치환의 결과는 (그림 5)와 같이 4 비트의 단위로 16개의 블록이 일정한 형태를 갖고 있다.

따라서 본 고에서는 16개의 블록을 각각 프로세서(혹은 프로세스)에 할당하여 병렬로 처리하여 DES 암호 알고리즘의 처리 속도를 향상시킨다.

치환 결과 값	M40 M8 M48 M16 M56 M24 M64 M32 M39 M7 M47 M15 M55 M23 M63 M31 M38 M6 M46 M14 M54 M22 M62 M30 M37 M5 M45 M13 M53 M21 M61 M29 M36 M4 M44 M12 M52 M20 M60 M28 M35 M3 M43 M11 M51 M19 M59 M27 M34 M2 M42 M10 M50 M18 M58 M26 M33 M1 M41 M9 M49 M17 M57 M25
할당 프로세서(프로세스)	P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16

(그림 5) 역 초기 치환의 병렬 처리

다. 확장 치환

확장 치환은 (그림 2)와 같은 연산을 16회의 반복 연산 과정에서 실행되는 부분으로 32 비트의 입력을 48 비트로 확장한 결과를 출력한다. 따라서 이 부분을 병렬로 처리하면 DES 암호 알고리즘의 처리 속도를 상당히 개선할 수 있다. 확장 치환의 결과는 (그림 6)와 같이 6 비트의 단위로 8개의 블록이 일정한 형태로 이루어져 있다.

따라서 본 고에서는 형태가 같은 8개의 블록을 각각 프로세서(혹은 프로세스)에 할당하여 병렬로 처리하여 DES 암호 알고리즘의 처리 속도를 향상시킨다.

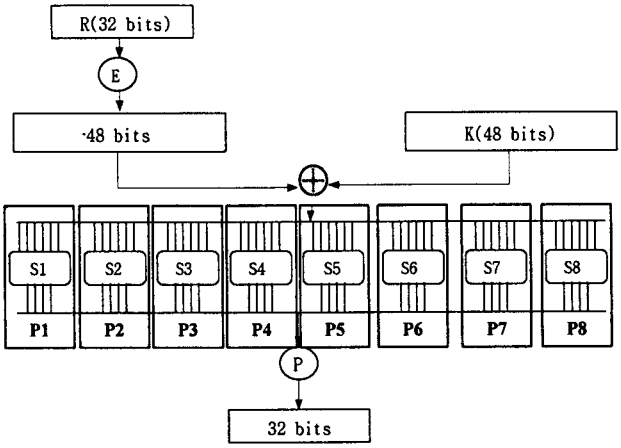
확장 치환 결과 값	블 록	할당 프로세서(프로세스)
M32 M1 M2 M3 M4 M5	블록 1	P1
M4 M5 M6 M7 M8 M9	블록 2	P2
M8 M9 M10 M11 M12 M13	블록 3	P3
M13 M14 M15 M16 M17 M18	블록 4	P4
M18 M19 M20 M21 M22 M23	블록 5	P5
M23 M24 M25 M26 M27 M28	블록 6	P6
M28 M29 M30 M31 M32 M33	블록 7	P7
M33 M34 M35 M36 M37 M38	블록 8	P8

(그림 6) 확장 치환의 병렬 처리

2. 대치 연산의 병렬 처리

DES 암호 알고리즘에서 대치 연산은 암호학적으로 가장 중요한 연산을 수행하는 부분이다. 대치 연산은 (그림 2)에서 보는 바와 같이 16회 반복 연산하는 부분에 속해있다. 따라서 대치 연산을 병렬로 처리하면 DES 암호 알고리즘의 처리 속도를 상당히 개선할 수 있다. 대치 연산은 특정 위치의 값을 다른 값으로 대치하는 연산이며, 대치 연산을 수행하기 위한 8개의 S-Box로 구성되어 있으며, 각각 6비트를 입력으로 받아 4비트의 출력 갖는다. 각 S-Box의 입력 6비트 중 첫번째 비트와 마지막 비트는 2비트의 조합으로 S-Box의 Row를 결정하고, 중간에 있는 4비트는 Column을 결정한다.

따라서 본 고에서는 대치 연산의 병렬 처리를 위하여 (그림 7)과 같이 S-Box 처리 단위로 각각 프로세서(혹은 프로세스)를 할당하여 DES 암호 알고리즘의 처리 속도를 개선하는 방법을 제시한다.



(그림 7) 대치 연산(S-Box)의 병렬 처리

IV. 결 론

DES 암호 알고리즘은 1977년 미국에서 표준 암호 방식으로 채택한 이후에 오늘날 까지 널리 소프트웨어 혹은 하드웨어 구현을 통해 사용하고 있다. 각종 통신 시스템, 컴퓨터 시스템에서 데이터 처리 중 암호화를 부가적으로 수행함에 따라 암호화 과정에서 병목현상 문제가 발생될 수 있다. 고속 컴퓨터 시스템, 고속 통신망 등의 응용 분야는 고속 암호화 처리가 필수적으로 해결되어야 될 과제이다. 현재 고속으로 정보를 보호하는 분야에서는 하드웨어로 구현된 DES 칩을 주로 이용하고 있으나 그 사용이 제한적이며, 하드웨어로 구현이 곤란한 분야는 소프트웨어로 구현하여야 하는데 소프트웨어로 DES 암호 알고리즘을 실현할 경우 고속의 실시간 처리에 문제가 있다.

따라서 본 고에서는 이러한 문제를 해결하기 위하여 DES 암호 알고리즘의 치환 연산 중에서 초기 치환 연산, 역 초기 치환 연산 그리고 확장 치환 연산의 병렬 처리 방법과 대치 연산(S-Box 처리)을 병렬로 처리하는 방법을 제시하였다. 본 고에서 제시한 방식을 이용하여 DES 알고리즘을 구현할 경우 DES 암호 알고리즘의 처리 속도를 상당히 향상시킬 수 있을 것이다.

[참고 문헌]

[1] NBS, Data Encryption Standard, *Federal information Processing Standard Pub. 46*, 1977.

[2] F. Hoornaert, J. Goubert, and Y. Desmedt, "Efficient hardware implementation of the DES," *CRYPTO'84*, 1984.

[3] I. Verbauwhede, F. Hoornaert, J. Vandewalle, and H. DeMan, "Security Considerations in the

Design and Implementation of a new DES chip," *Eurocrypt '87*, 1987.

[4] Hans Eberle,"A High-speed DES Implementation for Network Applications," *CRYPTO '92*, 1992.

[5] D. Williams and H. J. Hindin,"Can software do encryption job?," *Electronics*, 1980. 7.

[6] 박태규, 황대준, "DES의 고속 암호화를 위한 파이프라인 구조," *통신정보보호학회 논문지* 제 3권 제2호, 1993.