제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

EC에서 패스워드를 기반으로 한 병렬처리 사용자 암호해독 및 패스워드 관리에 관한 연구

정창렬*, 김단환**, 고진광*
*순천대학교 컴퓨터과학과, **조선대학교 컴퓨터공학과

The Parallel Processing User Decryption and User Password Management based the Password in Electronic Commerce

Chang-Ryul Jung*, Dan-Hwan, Kim**, Jin-Gwang Koh*
*Dept of Computer Science, Sunchon National University
**Dept of Computer Engineering, Chosun University

요 약

전자상거래의 인구는 매년 급속히 증가하고 있으며, 또한 전자상거래의 대다수 쇼핑몰 사이트가 패스워드를 기반으로 사용자를 인증하고 있다. 그런데 사용자는 이런 패스워드 기반 사이트를 방문하면서 보안과 안전을 고려하지 않고 패스워드를 만들어서 사용하고 있다. 이러한 패스워드는 사용자의 프라이버시의 침해와 개인정보가 노출이 되는 문제를 안고 있다. 이러한 문제점을 관리적 측면에서 패스워드를 해독하여 해독하기 쉬운 일반적이고 평이한 패스워드를 사용자는 mail를 통해 알려서 패스워드의 위험성을 주지시키도록 한다. 사용자의 패스워드를 알기 위해서는 암호해독 기법이 필요하는데 이 해독기법을 빠르고 정확하게 하기 위해서 분산화 된 동적 작업배분방법을 이용한 병렬처리 패스워드 해독 기법을 제안하여 구현하였다. 본 연구에서는 이러한 암호해독을 하여 전자상거래에서 사용자가 사용한 패스워드를 안전하게 관리할 수 있도록 하고, 사용자의 프라이버시를 효과적으로 보호 할 수 있는 모델을 제안한다.

1. 서 론

정보통신의 급속한 발전으로 인해 인터넷이용자 또한 크게 증가하였다. 인터넷이용은 국경을 초월한 새로운 교역시장으로 이용되는 전자상거래가 빠르게 성장하여 발전하고 있다.

인터넷을 이용한 전자상거래는 거래 유형에 따라 B2B, B2C, B2G, C2C, C2G등의 유형으로 구분한다. 전자상거래의 거래유형의 가장 대표적인 것은 B2B, B2G 그리고 B2C이다. 이는 통계청 발표에 의하면 지난해 우리나라 전자상거래규모 11조 9천 800억원에 이르며, 전년도에 비해 2배가 성장한 규모라고 라는 것이다. 거래유형에 따라 B2B가 108조9천억으로 거래 비율의 91.6%을 차지했고, B2G는 약 7조 370억원으로 거래비율의 5.9%를 차지했다. B2C는 2조 5천800억원으로 2.2%를 차지한다고 발표했다. 특히 B2C의 경우는 국내에서도 소액규모지만 많은 거래가 이루어지고 있음을 알 수 있다.

인터넷을 통한 전자상거래는 거래의 비용절감, 고객의 확대 등 여러 가지 이점을 제공하지만 먼저 선행조건으로 해결이 되어야 하는 문제가 있다[1].

B2C거래를 위한 대부분의 전자상거래쇼핑몰에서는 소규모로 이루어지고 있고, 사용자의 인증을 대부분 패스워드를 기반으로 한 사용자인증을 하기 때문에 일부 사용자는 패스워드를 일반적 또는 특정한 숫자나 낱말을 이용하여 사용[3]하기 때문에 자신의 프라이버시를 침해당하는 경우가 발생한다. 이러한 문제는 전자상거래 이용자에게 매우 심각한 문제 발생과 전자상거래가 발전하는데 저해요인으로 작용된다.

따라서 본 연구에서는 이러한 문제해결을 위해 사용자 패스워드를 해독하는 알고리즘을 적용하여 분산화된 동적작업배분방법을 이용 병렬처리 암호 해독기를 구현한다. 뿐만 아니라 암호해독을 하여 전자상거래에서 사용자가 사용한 패스워드를 안전하게 관리할 수 있도록 하고, 사용자의 프라이버시를 효과적으로 보호할 수 있는 모델을 제시한다.

본 연구의 구성은 제 2장은 병렬처리시스템의 구조와 분류에 대해 기술하고 제3장에서는 패스워드 기반 암호알고리즘, 제4장에서는 병렬처리 사용자 암호해독 및 사용자패스워드 관리모델 그리고 제5장은 결론을 기술 한다.

2. 병렬처리 시스템의 구조와 분류

병렬처리는 계산 속도를 높이기 위해 동시에 여러 작업을 처리하는 것, 또는 하나 이상의 연산을 동시에 수행하여 연산속도를 증가시키려는 처리방법으로서 최근 컴퓨터는 정도의 차이가 있을 뿐 모두 병렬처리능력을 가지고 있다.[8] 한 작업을 수행하는데 하나의 컴퓨터로 실행하는 것 보다 n개의 컴퓨터로 동시에 처리를 이용한다면 처리속도는 n배 증가 시킬 수 있다[2]. 이러한 병렬처리컴퓨터의 목적은 단위시간당 수행한 작업의 양 및 처리능력(throughpput)을 빠르게 향상시키는데 있다.

병렬처리시스템는 몇가지 방법에 의해 분류가 되는데 주요 분류방법은 동시에 처리할 수 있는 명령어나 데이터 수 , 처리기의 내부조직, 처리시간의 연결구조, 또는 시스템을 통하는 명령어와 데이터와 흐름을 제어하는 방법 등에 따라 분류를 한다. 대표적인 분류방법은 Flynn에 의한 분류가 있다. Flynn에 의한 분류는 컴퓨터 구조를 명령어 스트림과

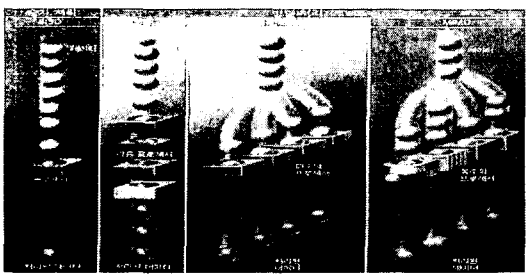

그림 1. 순차처리와 병렬처리 구조

데이터 스트림이 컴퓨터 내에서 각각 하나 또는 여러 개인가를 기준으로 분류하였는데, 하나의 데이터 스트림이라고 하면 하나의 명령어 스트림에서 요구되는 일련의 연속적인 데이터나 중간 결과를 의미하여 SISD, SIMD, MISD, MIMD로 분류하였다[2].

표 1 Flynn에 분류방식에 의한 컴퓨터시스템의 분류

Arch.	Computer
SISD	IBM701, IBM1601, IBM7090, PDP, VAX11
SISD	IBM S/360-91, IBMs/3701-68up, CDCD6600, DC Star-00, TI-ASC, FPS AP-120b, Cray-1, CDC Cyber-205, CDC-Nasf, Fujitsu FACOM-230/75
SIMD	ILLIAC-IV, PEPE, BSP
MIMD	IBM S/370-168mp, UNIVAC 1100/80, Tandem 16, Ibm 3081/3084, C.M
MIMD	Burroughs D-825, G.mmp, Cray-2, S-1, Cray-XMP, Denelcor HEP

MIMD는 병렬처리의 구성방법에 따라 MPMD와 SPMD로 구분되어지는데 SPMD는 작업을 지시하는 프로그램과 실제 작업을 수행하는 프로그램이 하나의 프로그램에 같이 쓰여지고 모든 프로세서가 같은 프로그램을 수행하도록 구성하는 방법이다. 하나의 프로세서가 작업을 지시하는 역할과 작업을 수행하는 역할을 모두 할 수 있기 때문에 프로세서를 계층적으로 구성할 때 효율적으로 사용될 수 있는 방법이다.

2.1 효율적인 작업배분 과 프로세서의 병렬화

2.1.1 동적 작업 배분

효율적인 작업배분은 정적 작업배분과 동적 작업배분으로 나누어지는데 동적 작업배분은 프로세서가 시작한 후 실행상황에 따라 작업을 배분하는 방식으로 앞으로 수행되어야 할 부분작업(work pool)을 프로세서가 실행되는 동안 유지관리하게 되며, 프로세서들이 작업이 필요할 때마다 유지 관리되는 부분작업들 중 하나를 배분함으로써 원래 주어진 작업이 완료 할 때까지 모든 프로세서들이 가능한 한 계속해서 작업을 수행할 수 있도록 하기 위한 방식이다.[9] 이러한 동적 작업배분은 집중화된, 분산화된, 완전 분산화된 동적작업배분으로 나누어 지는데 본 연구에 적용된 분산화된 동적 작업배분이다. 앞으로 수행되어야 할 부분 작업들이 계층화된 여러 프로세서들에게 나누어져서 유지관리 되는 방식으로 각각의 부분 작업 유지관리 프로세서들은 자신의 하위 프로세서들에게 작업을 배분하게 된다. 배분된 작업의 크기 정도는

$$\frac{\text{실제작업을 수행하는데 소요되는 시간}}{\text{작업배분을 위해 소요되는 시간}}$$

으로 나타낸다.

2.1.2 프로세서내의 병렬화

상위프로세서로부터 작업 지시를 받아 실제 작업을 수행하는 역할과 하위 프로세서들에게 작업을 지시하는 역할을 모두 실행할 수 있는 SPMD방식을 적용함으로서 동시에 여러 프로세서들과 연결이 이루어져 있을 수 있다. 병렬화함으로서 연결된 상대 프로세서들에 대한 응답시간을 줄일 수 있어 전체의 성능을 높일 수 있다. 이때 스레드를 이용하면 스레드는 프로세스의 실행환경을 공유하지만 독립적으로 실행될 수 있는 프로세스내의 하나의 제어흐름으로 하나의 연결에 대한 처리를 하나의 스레드가 맡도록 함으로써 여러 연결에 대한 처리를 병렬화 할 수 있게 된다.[9]

3. 패스워드기반 암호 알고리즘

패스워드를 기반으로 하는 암호화 알고리즘은 가장 전통적인 개인을 식별하는 방식의 알고리즘으로 사용되는 방식이다.[3] 패스워드를 기반으로 하는 암호 해독하는 방법은 크게 두 가지로 나눌 수 있는데 암호화 알고리즘 전체를 분석하여 암호 해독하는 방법과 모든 가능한 패스워드를 하나씩 체크하여 암호를 해독하는 방법 등이 있다[7]. 두 번째 방법은 알고리즘에 전문적인 지식이 없어도 가능한 방식으로 모든 패스워드를 해독하고자하는 암호문을 비교하여 일치하는 경우를 찾게 되는 데 이 모든 경우수를 시도하기란 엄청난 시간이 필요하다. 8비트의 패스워드를 생성하는데 128비트의 아스키코드라면 $128^8 = (2^7)^8 = 2^{56} \fallingdotseq 7.2 \times 10^{16}$ 기 된다. $1\mu s$ 의 프로세서로 8자리 패스워드를 해독한다면 2300년이 걸릴 것이다. 본 연구는 MD5를 기반으로 하는 패스워드 기반의 암호화 방법을 적용하여 최대 127문자까지 패스워드로 사용할 수 있고, 암호화된 패스워드의 길이는 34문자이며 그중 맨 앞문자 12문자는 salt이다.

패스워드가 'student'라고 할때 * "/ete/shadow " 파일에 기록되는 암호화된 패스워드

1ux6fwc. . . . $jscgcur6rc1* Salt : $1SUX7FWC . . . . . $ 패스워드를 암호화하기 위해 사용하는 Crypt()함수 프로토타입은 char * crypt(const char *key, const char *salt); 여기서 key는 사용자가 입력한 패스워드이며, 암호화된 패스워드가 원래의 패스워드로 변환된다.

4. 병렬처리 사용자 암호 해독 및 사용자 패스워드 관리 모델

병렬처리를 위한 시스템 구현 환경은 PentiumIII 800Mhz, RAM 128, LAN 3COM 100M, HUM는 Level one 8Port 10/100 switching HUB, C++ 프로그램 그리고 OS는 RedHat Linux 7.1 Kernel2.4.2이다. 하나의 자료를 동시에 적용하는 시스템의 SPMD를 방식을 적용하는 병렬처리이므로 작업지시를 하는 request()함수와 작업지시를 받아 실제 작업을 수행하는 receive_msg()함수까지 모두 하나의 프로그램d에 포함되어 있다. 각 프로세서는 계층형 트리 구조로 이루어져 임의 개수 하위프로세서들을 가질 수 있으며 연결은 process.list을 참조하여 이루어진다.

작업이 이루어지는 P_0, P_1, P_2 모든 프로세서는 하나이상의 작업수행 프로세스를 실행하여 업무를 수행함으로 실제로 적은수의 프로세서들로 효율적인 병렬시스템을 구축할 수 있다.

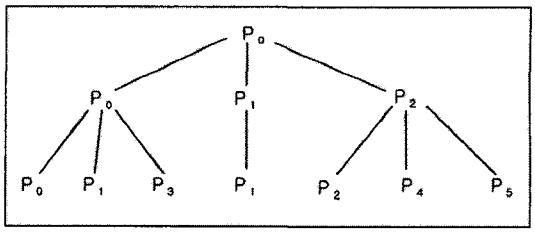

그림 2. 프로세스의 연결구조 예

패스워드가 알파벳으로 이루어져 있고 최대 길이가 5자리인 프로세스의 구성의 연결구조는 <그림 2>와 같이 트리구조로 나타내면 P_0 는 우선 길이가 하나인 패스워드를 하나씩 찾아 작업목록에 저장하고 없으면 끝까지 하위프로세스의 작업을 수행한다. P_2 는 길이가 2인 패스워드를 찾아 검사하고, P_5인 경우는 길이가 3에서 5이하인 모든 패스워드를 검사한다. 작업중인 프로세스가 종료하면 하위 프로세스에게 종료할 것을 지시한다. 암호해독을 위한 패스워드는 작업공간이 여러 작은 부분으로 쉽게 나뉘어지며 나누어진 작은 부분은 완전한 독립으로 탐색되어질 수 있음으로 병렬화 가능성(embarrassingly Parallel)이 매우 높다. 만약 트리구조의 depth=3이면 한 단계의 하위 프로세스들에게 더욱 세분화된 작업을 배분되었을 것이고, 소요되는 시간도 증가 되었을 것이다.

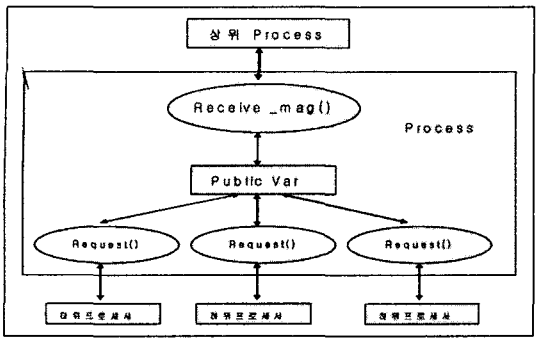

그림 3. 프로세서내부구조

<그림3>의 구조는 상위프로세스와 연결은 receive_msg()함수를 수행하는 스레드가, 하위 프로세스들과의 연결은 request()함수를 실행하는 스레드가 담당하여 스레드간의 정보교환은 Public Var 또는 신호를 이용하여 이루어진다[5].

이렇게 해서 얻어지는 결과는 최대 패스워드길이 가 5이고 12개의 알파벳 문자로 만들어진 모든 패스워드를 암호해독한 결과는 다음 표 2와 같다.

표 2 병렬 시스템의 실행 소요시간 비교

병렬시스템의 컴퓨터수	실행 소요시간
1	7분 7초
2	3분 33초
3	2분 22초
4	1분 46초
5	1분 47초

이렇게 해서 해독이 되어진 패스워드는 불안전성검사를 password policy list에 따라 검사를 하여 불안전성 판명을 받은 패스워드는 사용자의 ID와 함께 고객 데이터베이스와 연결하여 고객에게 보안의

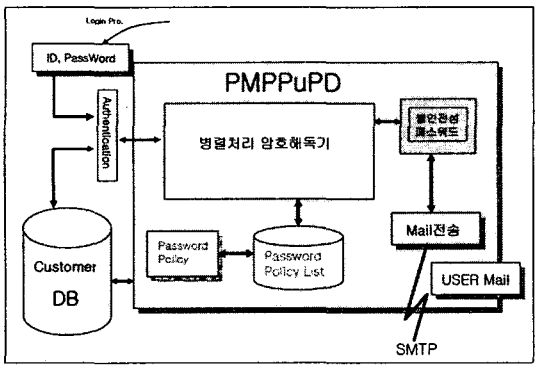

그림 3. 사용자 패스워드 관리 메커니즘

위험성을 알리는 e-mail을 통해서 DM를 발송한다.

<그림 4>의 구조는 병렬처리 병렬처리 암호해독 및 사용자 패스워드 관리 메커니즘을 나타내고 있다.

5. 결 론

패스워드를 기반으로 하는 전자상거래가 많이 보편화 되어 있으나 사용자의 개인프라이버시가 침해당하는 우려가 항상 배제하고 있다. 그러나 사용자의 패스워드를 불안전성을 체크하여 관리를 함으로써 개인 정보의 안전성을 높일 수 있을 것이다. 본 연구의 병렬처리 사용자 암호 해독은 순차프로그램으로 실행하였을 경우 7분 7초가 걸렸다. 그러나 병렬컴퓨터 수를 5대로 늘려 실행을 하면 1분 47초의 실행시간이 소요되었다. 컴퓨터 대 수와 소요시간과는 반비례함을 알 수 있다. 빠른 시간에 사용자의 패스워드 안전성을 체크하여 알려 줄 수 있다. 향후연구로는 여러 개의 암호화된 패스워드를 한꺼번에 해독하고 사용자가 바로 인증과 연결시키는 메커니즘의 연구가 계속되어야 할 것이다.

참고문헌

[1] 김홍근, 최영철, "전자상거래 정보보호기술 현황 및 대응방안", 한국정보처리학회지 6권1호, 1999.

[2] 조정완, 손진곤,「컴퓨터구조」, 한국방송대학교 출판부, 2001.

[3] 박창섭,「암호이론과 보안」, 대영사, 1999.

[4] 이상경, 조창열, "PC 클러스터를 이용한 수치최적설계의 병렬처리", 울산대학교 공학연구논문집 제33권 2호, pp.85-95, 2001.

[5] 김화중,「컴퓨터 네트워크 프로그래밍」, 홍릉과학출판사, 2000.

[6] 박두순, 이광영, 황종선, 김병수, "병렬처리를 위한 동기화 기법", 정보고학회지 제 13권 제7호, pp.132-140, 1995.

[7] 육군사관학교수학과,「암호학 개론」, 경문사, 1999.

[8] 한상영, 최영근, 원영주, "대규모 병렬처리 컴퓨터의 핵심 기술", 정보과학회지 제 13권 7호, p.16, 1995.

[9] SP Parallel Programming Workshop, "Parallel Programming Introduction"
http://www.mhpcc.edu/training/workshop