분산 컴퓨팅 환경에서 효율적인 암호 키 탐색 기법[1)]

이창호*, 강주성*[2)], 박태훈*, 최장원**
*국민대학교 수학과
**한국과학기술정보연구원 고성능연구망사업단
e-mail: jskang@kookmin.ac.kr

An Efficient Key Searching Method on Distributed Computing Networks

Chang-Ho Lee*, Ju-Sung Kang*, Tae-Hoon Park*, Jang-Won Choi**
*Dept of Mathematics, Kookmin University
**HPcN Project Division, KISTI

요 약

초고속 인터넷망이 발달됨으로써 분산 컴퓨팅 시스템 구축이 용이해졌다. 분산 컴퓨팅 시스템은 저비용과 유휴 계산 자원의 활용으로 기존의 슈퍼컴퓨터와 유사한 능력을 발휘할 수 있다는 장점을 지닌다. 암호 알고리즘의 실질적인 안전성 요소인 키의 길이는 전수조사 계산량에 의존한다. 키 전수조사를 위한 대용량 계산은 슈퍼컴퓨터, 클러스터, 분산 컴퓨팅 등의 환경에 따라 세부적인 메커니즘에 차이를 보인다. 본 논문에서는 분산 컴퓨팅 시스템을 소개하고, 이러한 환경 하에서 암호 알고리즘의 키 전수조사 작업을 수행하기 위한 세부적인 절차에 대해서 논하고, 구체적으로 키 전수조사 작업을 효율적으로 수행하기 위한 방법을 제안한다.

1. 서론

분산 컴퓨팅은 대용량 계산을 요하는 문제를 수많은 개별 컴퓨터가 병렬적으로 계산 가능한 부분으로 분할하여 각 개체의 결과를 통합함으로써 원래의 문제를 해결하고자 하는 계산 처리의 한 방법이다. 암호 알고리즘의 가장 기본적인 안전성 요소는 키 길이이다. 그리고 키 길이를 결정하는 실질적인 요소는 현재의 컴퓨팅 환경을 고려한 상대적인 공격량이다. 1970년대 중반에 미국 표준으로 제정된 DES[1]가 56-비트 길이의 키를 사용한 반면, 2000년도에 새로운 표준으로 선정된 AES[2]가 128-비트 이상의 키 길이를 재원으로 정한 중요한 요인은 동시대의 컴퓨팅 능력이다. 분산 컴퓨팅 환경은 이러한 암호 알고리즘의 키 길이를 실증적으로 평가할 수 있는 좋은 방법이다. 더욱이 PC의 계산 능력이 증가하고, 인터넷을 통하여 PC들이 손쉽게 연결될 수 있는 현재의 네트워크 환경에서 암호 알고리즘의 키 길이에 대한 안전성 평가 방법 중 가장 중요한 것이 바로 분산 컴퓨팅 환경에서의 공격 복잡도인 것이다.

본 논문에서는 분산 컴퓨팅 시스템을 소개하고, 이러한 환경 하에서 암호 알고리즘의 키 전수조사 작업을 수행하기 위한 세부적인 절차에 대해서 논하고, 구체적으로 키 전수조사 작업을 효율적으로 수행하기 위한 방법을 제안한다.

1) 본 지식재산권은 정통부 및 정보통신연구진흥원의 지원을 받아 수행된 연구 결과임(07-기반-04, 컴퓨터연계활용기반구축)
2) 교신저자

2. 키 전수조사와 계산 환경

암호 알고리즘의 실증적인 공격 복잡도는 비밀 키를 전수조사할 때 필요로 하는 계산량을 의미한다. 키 전수조사 공격은 올바른 키를 찾을 때까지 가능한 모든 키를 조사해 보는 암호 분석의 가장 기본적인 기술이다. 올바른 키를 식별하기 위해서는 평문과 대응하는 암호문 쌍이 필요하지만, 평문이 의미 있는 문자로 구성된 경우 암호문 단독으로도 올바른 키의 식별이 가능하다. 키 전수조사 공격은 임의의 암호 알고리즘에 대해서 적용이 가능하며, 키 스케쥴 또는 암호 알고리즘 자체의 약점을 이용할 경우 공격에 대한 효율성을 향상 시킬 수 있다.

키 전수조사 방법으로 키를 찾기 위해서는 매우 긴 시간을 필요로 한다. 미국 표준으로 사용되었던 DES를 예로 들어보자. DES의 경우 2^{56}개의 키가 존재하기 때문에 키 전수조사 방법으로 올바른 키를 찾기 위해서는 최대 2^{56}번의 암호화 작업을 필요로 한다. 이와 같은 작업은 1초에 200만개의 키를 체크할 수 있는 컴퓨터(펜티엄 4 2.66GHZ에서 1초에 약 170만개의 키를 체크함)가 1143년 동안 쉬지 않고 수행해야 하는 작업이다.

키 전수조사 공격과 같이 대용량의 계산을 요하는 작업을 위해서 과거에는 슈퍼컴퓨터가 사용되었다. 슈퍼컴퓨터는 특수한 계산 기법과 초고속의 계산 성능으로 다양한 분야의 대용량 계산이 가능하다는 장점을 지닌다. 하지만 슈퍼컴퓨터는 대단히 많은 구매 비용과 관리 및 유지 보수를 위해서 다수의 인력과 조직을 필요로 한다. 그러므로

슈퍼컴퓨터의 활용은 국가적으로 매우 중요한 분야에 그 용도가 제한될 수밖에 없는 실정이다.

슈퍼컴퓨터의 대안으로 키 전수조사 공격을 위해서 사용 가능한 것이 클러스터링 시스템이다. 클러스터링 시스템은 여러 대의 PC를 병렬로 연결하여 슈퍼컴퓨터에 준하는 계산 능력을 얻기 위해서 제안된 시스템이다. 슈퍼컴퓨터에 비해서 적은 비용으로 구축이 용이하다는 장점을 가지고 있지만, 클러스터링 시스템 구축에 필요한 공간과 비용은 여전히 부담스러운 상태이다. 또한, 확장성이 용이하다는 장점과 함께 확장에 필요한 PC의 구매 비용은 항상 추가적으로 발생한다는 단점을 안고 있다.

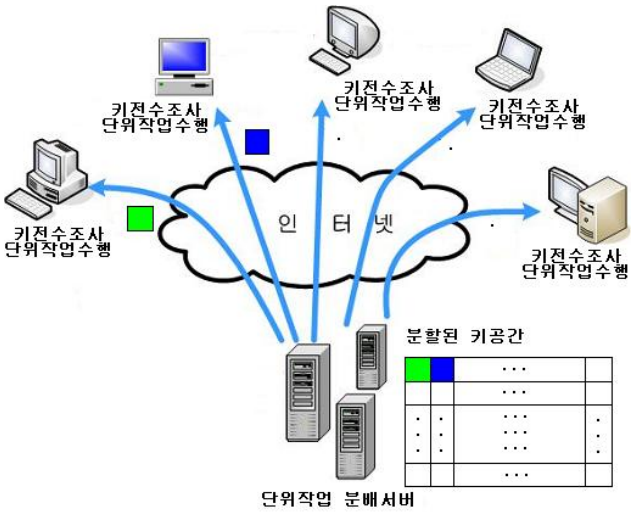
한편, 분산 컴퓨팅은 대용량 계산을 요하는 문제를 수많은 개별 컴퓨터가 병렬적으로 계산 가능한 부분으로 분할하여 각 개체의 결과를 통합함으로써 원래의 문제를 해결하고자 하는 계산 처리 방법이다. 키 전수조사를 위해서 필요로 하는 계산은 키 공간을 분할하여 병렬 계산이 가능하기 때문에 분산 컴퓨팅 시스템에 매우 적합하다. 개념적으로는 클러스터링 시스템과 크게 다를 바가 없지만, 물리적으로 연결되는 PC가 인터넷 등을 통하여 원격으로 접속할 수 있다는 점에서 차이가 있다. 개별 PC를 추가적으로 연결할 때 비용이 발생하지 않는다는 점과 유휴 자원을 활용할 수 있다는 점은 분산 컴퓨팅 환경이 갖는 큰 장점이다. 또한, 분산 컴퓨팅 환경에서는 네트워크로 연결되기만 하면 PC를 계산 자원으로 활용할 수 있기 때문에 확장성이 매우 용이하고 추가 비용이 소요되지 않는다[3].

분산 컴퓨팅 환경은 비교적 낮은 대역폭의 네트워크로 서로 연결되어 각자 다른 종류의 프로세서와 운영체제를 사용하여 독립적으로 운영되면서 메시지 전달을 통해 작업을 수행하는 환경을 말한다[4]. 다시 말해 계산량이 많은 작업을 원격의 컴퓨터들이 상대적으로 적은 양의 계산량으로 나눠서 수행하는 것이다. 현재 인터넷의 활성화와 함께 인터넷 기반 분산 컴퓨팅 환경이 많이 구축되어 있다.

3. 키 전수조사를 위한 분산 컴퓨팅 시스템

분산 컴퓨팅 시스템은 하나의 서버와 많은 수의 클라이언트로 구성되는데, 클라이언트에 에이전트 프로그램이 설치되어 서버와 통신을 하게 된다. 그러므로 앞으로 본 논문에서는 클라이언트와 에이전트를 구분하지 않고 에이전트로 통일하여 부르기로 한다. 즉, 통신을 통해 필요한 파일을 주고받으면서 서버와 에이전트가 각자의 작업을 수행한다. 일반적으로 분산 컴퓨팅 시스템에서의 대용량 계산은 통신량에 비해 계산량이 상대적으로 많을 때 사용된다[5].

암호 알고리즘의 키 전수조사 역시 계산량이 많은 작업이며, 키의 개수를 계산량으로 볼 수 있다. 이러한 경우 키 공간을 분할하여 단위작업으로 분리시킴으로써 계산량이 많은 작업을 개별 PC에 적은 계산량의 작업을 분배하는 것이 된다. 이와 같은 암호 알고리즘의 키 전수조사를 위한 분산 컴퓨팅 시스템의 예를 (그림 1)에서 볼 수 있다. 그림에서 보는 바와 같이 분할된 각각의 키 공간은 하나의 단위작업이 되어 순차적으로 에이전트에 보내지게 된다[6].



(그림 1) 분산 컴퓨팅 환경

3.1. 서버

서버는 하나의 큰 작업을 총괄해야 한다. 즉, 서버는 모든 작업과 에이전트들을 관리하며 에이전트와 통신하여 단위작업을 전송하고, 에이전트에서 완료한 결과를 받아야 한다. 제일 먼저 새로운 에이전트가 추가되면 서버는 필요한 파일을 에이전트에 전송하여 에이전트 프로그램을 설치하게 한다. 에이전트 프로그램이 실행되게 되면 서버는 실행파일과 몇 개의 단위작업을 에이전트에 전송한다. 그 뒤 서버는 필요한 작업을 수행하면서 대기하다가 단위작업을 요청하는 에이전트가 발생하면 다음 단위작업을 전송하게 된다. 하지만 종종 이러한 일련의 과정이 항상 지켜지는 것은 아니다. 서버가 에이전트에 단위작업은 전송했지만 결과가 돌아오지 않는 경우가 있을 수 있다. 개인 사용자가 에이전트 프로그램이 동작하는 중에 PC를 종료시켰을 때가 이러한 경우에 속한다. 이런 일을 해결하기 위해서 서버는 하나의 단위작업이 한 PC에서 수행되는데 걸리는 최대 시간의 예상 값을 알고 있어야 한다. 그리고 이를 바탕으로 하여 일정 시간 동안 수행 결과에 대한 응답이 없는 에이전트에 대한 단위작업을 다른 PC 즉, 다른 에이전트에 배포하도록 설정되어 있어야 한다.

3.2. 에이전트

에이전트는 단위작업에 참여하는 각 PC를 말한다. 에이전트 프로그램이 설치되어 서버와 통신하고, 작업을 요청하는 것을 통해 단위작업을 받아서 수행시키며, 그 결과를 서버에 되돌려 주는 역할을 한다. 에이전트 프로그램이 실행되면 에이전트는 제일 먼저 서버에게 PC의 CPU,

RAM 등의 하드웨어 정보를 송신하여 수행해야할 작업에 에이전트가 참여할 수 있는지 확인하게 된다. 에이전트가 단위작업의 수행에 참여하기 위해서는 서버가 요구하는 최소의 하드웨어 사양을 만족시켜야 한다. 또한, 에이전트는 각 PC에서 일정한 디렉토리를 자동으로 구성하여 실행파일과 단위작업, 그리고 결과파일의 위치를 정해준다.

4. 키 전수조사 작업의 효율적인 수행 방법

암호 알고리즘의 키 전수조사에서 일정량의 키를 체크하는 것은 하나의 단위작업을 수행하는 것을 의미한다. 따라서 효율적인 키 전수조사를 하기 위해서는 단위작업을 구성하는 세부적인 방법이 필요하다. 그리고 키를 체크할 때 키가 중복되지 않게 한번 씩 이루어져야 하고, 공백이 발생하면 안 된다. 그리고 키가 무엇인지 모르기 때문에 전체 키 공간에서 어떠한 방법으로 키를 체크해 나갈지 결정하는 것도 중요한 요소이다.

4.1. 에이전트 작업 시간의 효율적인 설정

에이전트에서 실행되는 단위작업의 실행 시간은 중요한 설정 사항이 된다. 단위작업의 실행 시간이 너무 짧으면 계산량에 비해 통신량이 증가되기 때문에 서버에 부하가 걸리게 된다. 반면 단위작업의 실행 시간이 너무 길면 서버가 단위작업을 관리하는데 문제가 생길 수 있다. 왜냐하면 앞에서 언급한 바와 같이 서버는 되돌아오지 않는 결과 값에 대한 단위작업을 다시 배포하는 일을 하기 때문이다. 이와 같은 이유로 단위작업의 실행 시간은 너무 짧지도 않고 길지도 않아야 한다. 실제로 운영되고 있는 분산 컴퓨팅 시스템인 Korea@Home[4]의 다양한 프로젝트에 의하면 경험적으로 하나의 단위작업 실행 시간은 15분 내외로 결정하는 것이 가장 적절한 것으로 되어 있다.

<표 1> 키의 개수에 따른 체크시간

키의 개수	소요 시간
2^{26}	36초
2^{27}	1분 14초
2^{28}	2분 30초
2^{29}	5분 11초
2^{30}	10분 32초

단위작업의 실행 시간을 적절하게 결정한 것을 바탕으로 분산 컴퓨팅 환경에서 DES의 키 전수조사 공격을 실시한다고 생각해보자. 먼저 단위작업의 실행 시간을 설정해 주어야 한다. 이를 위해 키의 개수에 따라 키를 체크하는데 걸리는 시간을 측정해 보았다. 이 시간은 암호 알고리즘의 암복호화 속도에 의존한다. 펜티엄 4 CPU 2.66GHz, 512MB RAM, 윈도우 XP의 PC사양에서 실험결과는 <표 1>과 같다.

단위작업을 수행할 때 최소 1블록에서 최대 7블록의 암호문을 복호화를 하게 된다. 즉, 평균적으로 4개의 블록을 복호화하여 2^{30}개의 키를 체크하게되는데 걸리는 시간은 약 13분이다. 따라서 하나의 단위작업이 2^{30}개의 키를 체크 할 수 있도록 실행파일과 입력파일을 구성하면 된다.

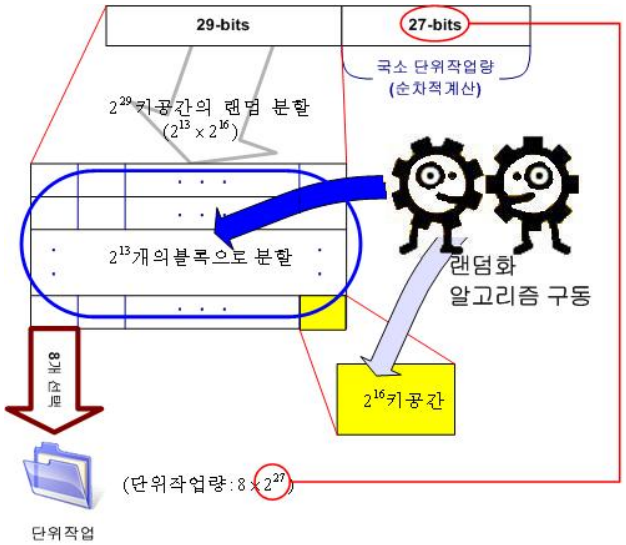
4.2. 효율적인 키 공간 분할 방법

이제 효율적인 키 공간 분할 방법에 대해 알아보자. 보통 키 전수조사를 할 때 난수를 생성하여 그 값이 키가 맞는지 체크를 하지만, 전체 키 공간을 확인하기 전에 중복되는 난수가 생성되는 경우가 발생한다. 예를 들면, 의사난수를 생성할 때도 약간이지만 특정 값에 편중되는 현상이 있다. 실제로 DES의 56-비트 키를 축소시켜 시뮬레이션 해본 결과 초기 값으로 난수를 생성하여 1씩 증가시키면서 키를 체크하는 방법이 난수만 사용하는 방법보다 빠르게 키를 찾는 것을 알 수 있었다. 이와 같은 결과를 통해 하나의 단위작업에 해당하는 2^{30}개의 키를 순차적이면서 랜덤성을 부여하기 위해, 시리얼하게 증가하는 구간은 2^{27}개로 하고, 2^{3}개의 시작점을 갖게 하였다[7][8]. 즉, 난수인 시작점을 입력받으면 시작점으로부터 1씩 증가시켜 2^{27}이 더해진 값까지 키를 체크하도록 실행파일을 작성하였고, 입력파일은 8개의 시작점을 포함하게 구성하였다.

분산 컴퓨팅 환경에서는 서버가 단위작업을 관리해야 한다. 즉, 키를 관리한다고 볼 수 있다. 서버가 찾아야 할 키를 에이전트에 보내기 때문에 서버가 단위작업을 랜덤한 순서로 분배가 가능하면 좋을 것이다. 랜덤성이 필요한 이유를 위에서 잠깐 이야기 했지만 키를 찾는 입장에서 랜덤성의 보장은 다른 이유가 포함된다. 큰 공간에서 키를 찾을 때 국소적으로 찾는 것이 아니라 전체적으로 골고루 찾는 것을 의미한다. 즉, 에이전트 입장에서 누구든지 동등한 키 탐색 성공 기회를 부여한다고 말할 수 있다. 랜덤성이 좋다는 것은 키 공간 전체가 키를 찾을 확률이 같다는 것이다. 이러한 이유로 서버에서 입력파일들을 랜덤한 순서로 분배가 가능하면 좋을 것이다. 하지만 이는 관리 측면과 구현의 측면에서 쉽지 않은 문제를 야기한다. 대부분의 서버는 자료들을 순차적으로 분배하도록 구축되어있다. 이러한 환경에서 랜덤성을 보장하기 위해서는 입력파일을 만들 때 시작점들의 순서 자체를 랜덤하게 하여 만들어야 한다.

입력 파일에 랜덤성을 부여하여 단위작업을 구성하는 방법을 다음과 같이 제안한다. 먼저 시리얼하게 체크하는 크기인 2^{27}로 DES의 전체 키 공간을 나누면 2^{29}개의 시작점이 만들어진다. 2^{29}개의 시작점으로 만들어지는 입력파일은 약 7GB이다. 이러한 시작점을 배열에 넣을 수 있게 (그림 2)처럼 2^{13}개의 블록으로 분할한다. 그 결과 한 블록에는 2^{16}개의 시작점이 존재하게 된다. 먼저 블록들의 순서를 섞어준 뒤, 첫 번째 블록을 선택한다. 그 블록에 해당하는 2^{16}개의 시작점을 섞은 후, 8개씩 끊어서 입력파일을 만든다.

이와 같은 방법으로 두 번째 블록부터 2^{13} 번째 블록까지 수행하여 입력파일들이 랜덤성을 갖게 한다.



(그림 2) 단위작업 구성 방법

4.3. 효율적인 랜덤성 확보 알고리즘

입력파일들이 효율적인 랜덤성을 확보하도록 하기위해 2^{16}개의 배열을 이용해 몇 가지 실험을 실시하였다. 첫 번째 방법은 DES를 이용하여 키를 고정시키고, 배열의 자리 번호를 입력 값으로 하여 출력 값을 뽑아낸 다음, 그 출력 값을 자리 번호로 인식하여 해당 위치의 값을 교환하는 것이다. 두 번째 방법은 ANSI X9.17 의사난수 발생기[9]를 이용하여 두 개의 난수를 선택한 후, 그 값을 자리 번호로 하는 배열 값을 서로 교환하는 것이다. 각각의 방법에 대해 중간 과정을 각각 2^{15}번, 2^{16}번씩 수행하게 하여 총 4가지 실험을 100회 실시한 다음, 바뀌지 않은 배열의 개수를 계산해 보았다. 실험결과는 <표 2>와 같다.

<표 2> 배열 섞기

랜덤성 확보 방법	바뀌지 않은 배열의 개수(100회 평균)
DES(2^{15})	19882.28
DES(2^{16})	0.79
의사난수발생기(2^{15})	24118.86
의사난수발생기(2^{16})	8874.38

<표 2>에서 보는 바와 같이 첫 번째 방법을 모든 배열에 실행한 것이 거의 모든 배열의 위치를 변화시켰다. 따라서 이와 같은 방법으로 입력파일을 생성할 경우 랜덤성을 효과적으로 확보할 수 있다고 볼 수 있다. 더욱이 첫 번째 방법은 의사난수 발생기를 이용하는 두 번째 방법에 비해서 한층 더 효율적이기 때문에 랜덤성 확보를 위한 알고리즘으로 첫 번째 방법을 선택하는 것이 바람직하다.

5. 결론

분산 컴퓨팅 시스템은 저비용으로 유휴 계산 자원을 활용하여 기존의 슈퍼컴퓨터와 유사한 능력을 발휘할 수 있다는 장점을 지닌다. 암호 알고리즘의 실질적인 안전성 요소인 키의 길이는 전수조사 계산량에 의존하며, 키 전수조사를 위한 대용량 계산은 슈퍼컴퓨터, 클러스터, 분산 컴퓨팅 등의 환경에 따라 세부적인 메커니즘에 차이를 보인다. 우리는 분산 컴퓨팅 시스템 환경 하에서 암호 알고리즘의 키 전수조사 작업을 수행하기 위한 세부적인 절차에 대해서 논하였다. 구체적으로 키 전수조사 작업을 효율적으로 수행하기 위한 방법을 살펴보았다. 에이전트 단위작업 시간의 효율적인 설정 방법과 키 공간의 적절한 분할 방법 및 랜덤성 확보 알고리즘 등에 대해서 각종 실험을 통한 연구 결과를 제시하였다. 향후 좀 더 많은 실험을 통하여 안정성 높은 키 전수조사 메커니즘에 대한 연구가 진행되어야할 것으로 생각된다.

참고문헌

[1] FIPS 46, "Data Encryption Standard", U.S. Department of Commerce, 1981.

[2] FIPS PUB 197, "Announcing the Advanced Encryption Standard (AES) ", NIST , 2001

[3] http://www.distributedcomputing.info/

[4] http://www.koreaathome.org/

[5] http://www.distributed.net/

[6] A. P. L. Selkirk, A. E. Escott "Distributed Computing Attacks on Cryptographic Systems", BT Technology Journal, Vol. 17, No. 2, 1999.

[7] Dirk Koch, Matthias Korber, and Jurgen Teich, "Searching RC5-Keys with Distributed Reconfigurable Computing", http://ww1.ucmss.com/, 2006.

[8] Matt Curtin, Justin Dolske, "A Brute Force Search of DES Keyspace", 1997.

[9] ANSI X9.17, "American National Standard -Financial institution key management(wholesale)", ASC X9 Secretariat-American Bankers Association, 1985.