클라이언트/서버 모델을 이용한 키 전수조사 시스템 구현

송 상 헌† · 박 현 동† · 김 충 길† ·
박 중 길† · 김 영 진† · 류 재 철††

요 약

사용자의 정보를 보호하기 위해 일반적으로 암호 알고리즘을 이용하지만, 대부분의 암호 알고리즘은 공개되어 있기 때문에 안전성(Security)은 사용된 암호 키(Key)의 길이에 의존하게 된다.

이에 따라 암호 시스템에 대한 공격 방법 중 하나로써 키 공간을 모두 조사하여 암호문에 사용된 키 또는 평문을 알아내는 방법있는데 이를 "Brute Force Attack"이라고 한다. 최근 하드웨어의 수행속도와 컴퓨터를 이용한 분산처리 기술이 발전함에 따라 여러 대의 컴퓨터를 네트워크로 연결하고, 암호 알고리즘에서 사용되는 키 공간 전체를 나누어 조사하는 클라이언트/서버 시스템 구축이 시도되고 있다.

본 논문에서는 메시지의 암호화에 일반적으로 이용되고 있는 관용 암호 알고리즘의 다양한 키 길이(40~256 비트)에 대해서 키 공간(Key Space) 전체를 조사할 수 있도록 키 블록 할당 알고리즘을 설계하고, 암호화에 사용된 키를 찾는 클라이언트/서버 시스템을 구현하였다. 또한 이 시스템을 이용하여 RC4 40비트 키 길이를 찾는데 소요되는 시간 비용을 분석하였다.

The Implementation of a Brute Force Attack System using a Client/Server Model

Sang-Heon Song† · Hyun-Dong Park† · Chung-Kil Kim† ·
Jung-Gil Park† · Young-Jin Kim† · Jae-Cheol Ryou††

ABSTRACT

We use an encryption algorithm to protect our sensitive information. But most encryption algorithms are open, so the level of security depends on the length of key. Therefore, there is an attack called "Brute force attack" which checks all possible keys. Nowadays, the technology of hardware and distributed system has progressed. So, many people try to recover the ciphertext without key information using several computers connecting each other via network.

In this paper, In order to checks all possible key of conventional algorithms using a various key length that generally encrypts messages, we designed an algorithm which assigned key block to each computer and implemented a client/server system which is able to seach a key used in encryption algorithm. Moreover we analyzed the system performance factors including time and cost for finding 40 bits length of key of RC4 algorithm.

※ 본 연구는 한국과학재단 특정기초연구과제(과제번호:97-01-00-06-01-3) 연구비 지원에 의해 수행되었음.
† 준 회 원 : 충남대학교 대학원 컴퓨터과학과
†† 종신회원 : 충남대학교 컴퓨터과학과 교수
논문접수 : 1999년 1월 26일, 심사완료 : 1999년 4월 20일

1. 개 요

인터넷의 활용이 점차 일반화되어 감에 따라서 인터넷 사용자들은 개인 정보의 유출에 큰 우려를 나타내고 있다. 현재 이러한 문제점을 해결하기 위해서 인터넷을 통해 전송되는 중요한 정보를 제3자가 알아볼 수 없는 형태로 암호화하여 전송할 수 있도록 다양한 암호 제품이 개발되고 있다. 이러한 암호 제품에서 이용되는 암호시스템은 평문을 암호화의 동작을 거쳐 암호문를 생성하고, 역으로 암호문을 복호화를 통해 평문을 생성하는 메커니즘을 구현한 것이다.

이와 같은 암호 시스템은 암호화와 복호화를 수행하는 일종의 수학적 함수인 암호 알고리즘과 이 알고리즘의 동작에 필요한 비밀정보인 키로 구성되는데 일반적으로 암호 알고리즘은 수학적으로 안전성이 증명되어 공개된 상태로 사용되지만, 알고리즘의 동작에 부가적으로 사용되는 키는 비공개로 안전하게 보관하여 사용해야 한다. 또한 이러한 키는 쉽게 알아낼 수 없도록 하기 위해 암호 시스템마다 다양한 키 길이를 정의해서 사용하고 있는데, 이러한 암호 시스템 개념을 간단히 살펴보면 다음과 같다.

암호 시스템은 크게 관용 암호 시스템과 공개키 암호 시스템 두 가지로 분류할 수 있다. 전자는 암호화 및 복호화에 사용되는 키가 동일한 형태의 시스템으로써 DES, IDEA, RC2, RC4 등과 같은 다양한 알고리즘이 있다. 후자는 암호화에 사용되는 키와 복호화에 사용되는 키가 서로 다른 비밀키와 공개키 한 쌍을 이용하는 시스템으로써 RSA, ElGamal, LUC 등과 같은 알고리즘이 있다.

관용 암호 시스템은 공개키 암호 시스템에 비해 암호화와 복호화를 수행하는 속도가 빠른 장점이 있지만 메시지를 주고 받는 송수신자 사이에 하나의 키를 공유해야 하기 때문에 이것를 안전한게 분배하는데 어려움이 따른다. 반면에 공개키 암호 시스템은 연산 속도가 느린 단점을 지니고 있기는 하지만 사용자가 비밀키와 공개키를 구분하여 자신의 공개키는 메시지를 주고 받는 어떤 사용자에게도 공개할 수 있기 때문에 키 분배 문제를 쉽게 해결할 수 있다[1,2].

이러한 특징으로 인해서 일반적으로 관용 암호 시스템은 임의의 키를 생성하여 메시지를 암호화하는데 사용하고, 공개키 암호 시스템은 메시지의 암호화에 사용된 관용 암호 시스템의 키를 암호화하는데 이용한다. 네트워크를 통해 송신자 A와 수신자 B 가 메시지를 주고 받을 경우 두 가지의 암호 시스템은 (그림 1)에서 보는 바와 같이 혼용하여 사용된다[2].

송신자 (A)	ECK [메시지] || EPPKB [K]	수신자 (B)
EC : 관용 암호 알고리즘 EP : 공개키 암호 알고리즘		K : 세션키 PKB : B의 공개키

(그림 1) 관용 암호 알고리즘과 공개키 알고리즘의 혼용

먼저 송신자 A는 임의의 키를 생성하여 관용 암호 알고리즘을 이용해 메시지를 암호화한다. 이때 사용한 세션키는 수신자 B의 공개키로 암호화 한 후 암호화된 메시지와 함께 수신자B에게 전송해 주게 된다. 이를 수신한 B는 자신의 비밀키로 암호화된 키를 복호화하고, 이 키를 이용해 암호화된 메시지를 복호화함으로써 수신자 B는 메시지를 읽을 수 있게 된다. 이 경우에 제삼자는 B의 비밀키를 알지 못하므로 암호화에 사용된 키를 알 수 없고 메시지는 안전하다.

그러나 네트워크을 통해 전송되는 암호문은 노출되어 있어 이를 복호화하기 위한 키를 알아낼 수 있다면, 송신자 A와 수신자 B 사이의 메시지 교환은 암호 시스템을 사용한다 할지라도 안전하다고 할 수 없다. 현재 대부분의 암호 알고리즘은 공개되어 사용되는데 알고리즘 자체는 안전하다고 가정할 경우 암호 시스템의 안전성은 암호화를 수행할 때 사용된 키의 길이에 의존하게 된다[1]. 그러나 키의 길이에 의존적인 암호 시스템은 하드웨어 수행속도와 분산처리 기술이 발전함에 따라 여러 대의 컴퓨터를 네트워크로 연결하여 모든 키 공간을 조사해 봄으로써 암호문을 만드는데 사용된 키와 평문을 찾아 내는 방법이 가능하게 되었다[3].

이러한 방법을 이용한 시스템은 1997년 1월에 시작한 "RSA Secret-Key Challenge Contest"를 계기로 활발하게 개발 중에 있다. 이 콘테스트는 암호문과 암호화에 사용된 알고리즘 및 키 길이를 알려주고, 암호화에 사용된 키를 찾아 내는 것인데, 총 13개의 콘테스트 중 DES 56비트, RC5 40비트, 48비트, 56비트 암호키 찾기는 이미 종료되었으며, 나머지 RC5 64,72, 80, 88, 96, 104, 112, 120, 128비트 암호키 찾기가 현재 진행 중에 있다. RSA사는 이 콘테스트를 위해 총 116,000

달러의 상금을 제공하고 있다.

현재 암호키 찾기가 종료된 4개의 콘테스트를 수행하는데 걸린 시간을 살펴보면 <표 1>과 같다[4]. 각 수행 시간은 키 공간을 모두 조사하여 걸린 시간은 아니지만 56비트 이하의 키 길이에 대해서는 비교적 단시간 내에 암호키를 찾아 낼 수 있음을 알 수 있다. DES 56비트의 키를 찾는데 1일 평균 연결된 클라이언트의 수는 14,000대 정도였으며, RC5 56비트의 경우는 정확한 클라이언트 수는 언급하고 있지 않지만 Pentium Pro 200Mhz 프로세서 14,685개가 동작하는 연산 속도였다고 밝히고 있다.

〈표 1〉 완료된 "RSA Secret-Key Challenge Contest" 수행 시간

키 탐색	DES 56비트	RC5 40비트	RC5 48비트	RC5 56비트
수행시간	140일	3.5시간	313시간 (약 13일)	265일

아직 국내에서는 이와 같은 연구가 미진한 가운데 본 논문에서는 관용 암호 알고리즘에서 사용하는 40~256비트 사이의 다양한 키 길이에 대해서 키 공간 전체를 조사하여 암호화에 사용된 키를 찾는 클라이언트/서버 시스템을 구현하였다. 개발된 시스템을 이용하여 RC4/40비트 키를 찾는 작업을 시험적으로 수행하고, 이후에는 RSA사 콘테스트에 참여할 예정이다.

2. 키 길이에 따른 안전성 고찰

일반적으로 메시지의 암호화에 사용되는 관용 암호 알고리즘은 <표 2>에서 보는 바와 같이 다양한 알고리즘이 존재하는데 각기 다른 키 길이를 사용하고 있음을 알 수 있다[1].

암호 시스템에서 가능한 키 값의 범위 즉, 키 공간은 40비트 크기의 키 길이 경우에는 가능한 키의 개수가 2^{40}(1,099,511,627,776)개가 존재하게 된다. 결국 40비트 키를 이용해서 메시지를 암호화할 경우 1,099,511,627,776개의 가능한 키 중에서 하나의 키를 사용할 수밖에 없음을 의미한다. 이와 같이 제한된 키의 수로 인해 암호문을 해독하기 위해서 모든 키를 조사해 봄으로써 암호화에 사용한 키와 평문을 알아내는 것이 가능한데, 이러한 방법을 이용한 공격을 "Brute Force Attack"이라고 한다.

〈표 2〉 관용 암호 알고리즘에서 사용하는 키 길이

암호 알고리즘	키 길이 (비트)
DES	56
LOKI, FEAL, CAST	64
Skipjack	80
Khafre, SAFER	64 또는128
IDEA, Lucifer, MMB	128
REDOC	160
GOST	256
Khufu	512
RC2, RC4, RC5, Blowfish	다양한 키 길이 사용 가능

최근 이러한 공격을 시도하기 위해서 인터넷 사용자들의 PC, 프로그램이 가능하고 수행속도가 빠른 칩(Chip)으로써 FPGA(Field Programmable Gate Arrays)과 ASIC(Application-Specific Integrated Circuits) 등을 결합한 시스템들을 네트워크상에서 서로 연결하여 키를 찾는 연구가 진행되고 있다[4].

FPGA를 이용할 경우 초당 3만개의 키를 조사할 수 있으며, ASIC를 이용할 경우는 초당 2억개의 키를 조사할 수 있다. 이처럼 최근 값싸고 빠른 하드웨어 수행속도로 인해 과거에는 거의 불가능해 보였던 "Brute Force Attack" 방식은 상당한 의미를 갖는 공격 방법이 되었다. <표 3>은 관용 암호 알고리즘에 대해서 분석한 "Brute Force Attack"에 소요되는 시간을 나타내고 있는데, 40비트의 키 길이는 0.0002초에서 최대1주일의 시간이면 키를 찾아 낼 수 있으며, 56비트의 키 길이는 12초에서 38년 정도의 시간이 소요됨을 나타내고 있다. 또한 키의 길이는 길수록 전수 조사를 통해 키를 찾아 내는 것이 어려움을 알 수 있는데, <표 3>은 전수 조사 참여자에 따라 "Brute Force Attack"에

〈표 3〉 40비트, 56비트 키 길이에 대한 Brute Force Attack에 소요되는 시간

참 여 자	비 용	시스템	키 탐색 소요 시 간		요구되는 최소 키 길이(비트), 1995년 말
			40비트	56비트	
해 커	소액	PC	1주일	불가능	45
소 기 업	$400 $1,000	FPGA FPGA	5시간 12분	38년 556일	50 55
중 기 업	$300,000	FPGA ASIC	24초 18초	19일 3시간	60
대 기 업	$10,000,000	FPGA ASIC	7초 0.005초	13시간 6분	70
연 구 소, 국가기관	$300,000,000	ASIC	0.0002초	12초	75

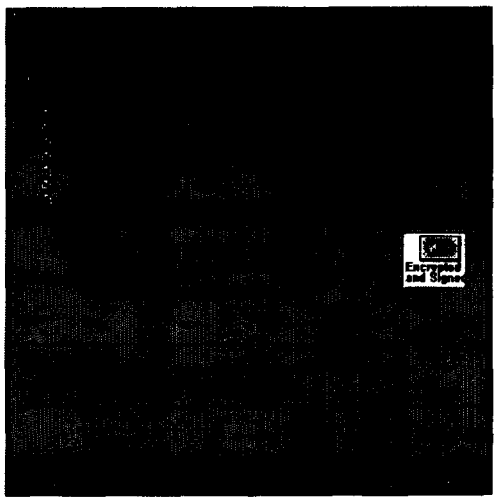
대응할 수 있는 최소 키의 길이는 45~75비트 정도가 요구됨을 나타내고 있다[3,5,6].

따라서 45비트 이하의 키 길이를 사용하는 관용 암호 알고리즘을 메시지의 기밀성 제공을 목적으로 이용하는 것은 부적합함을 알 수 있다. 즉, 추가적인 비용 없이 크랙커(Cracker)에 의해서 쉽게 공격될 수 있다는 것이다. 그럼에도 불구하고 현재 국내에서는 40비트 키를 사용하고 있는 외국 암호 시스템을 널리 이용하고 있다.

이러한 암호 제품의 대표적인 예는 Netscape사와 MS사에서 제공하고 있는 전자우편 클라이언트와 웹 브라우저(Web Browser)를 들 수가 있다((그림 3) 참조). 미국 내에서 사용하고 있는 이들 제품들은 128비트 키를 사용하고 있지만, 미국 이외에서 판매되는 제품은 암호 제품에 대한 수출 규제를 통해 40비트의 키만을 사용하도록 하고 있다[7,8,9]. 따라서 국내에서 이와 같은 외국 암호 제품을 사용하여 인터넷 서비스를 제공하는 것은 안전하다고 말할 수 없는 상황이다. 특히, (그림 2)에서 보는 바와 같이 국내 쇼핑몰의 대부분의 경우에 SSL(Secure Socket Layer)를 이용하여 신용카드와 같은 지불정보를 전송하고 있는데, 이 것은 매우 위험한 일이라 할 수 있다.

(그림 2) SSL 기반의 웹 브라우저

이에 따라 본 논문에서는 40비트 키를 사용한 암호 시스템의 위험성을 보여주기 위해 클라이언트/서버 기반의 키 전수 조사 시스템을 제시하고자 한다.



(그림 3) S/MIME 기반의 전자우편 클라이언트

3. 키 공간 탐색을 위한 클라이언트/서버 시스템 설계

관용 암호 알고리즘에서 이용되는 다양한 길이의 키를 찾는데 키 전수 조사 방법으로 키를 찾을 경우 암호 알고리즘마다 차이가 있기는 하지만 많은 수행시간이 요구된다.

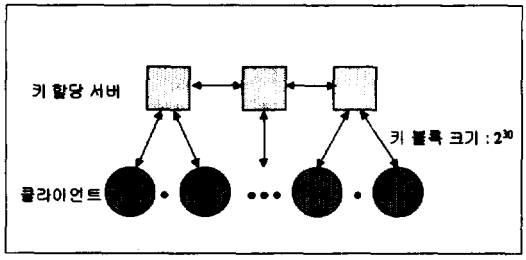
따라서 수행시간을 단축하기 위해서 키 공간을 블록 단위로 분할하여 다수의 클라이언트에 키를 할당해 탐색하게 하는 클라이언트/서버 모델을 도입하였다. 서버는 키 공간을 분할하여 클라이언트에게 할당하고, 클라이언트는 자신이 할당받은 키를 조사하여 결과를 알려주는 역할을 한다. 여기에서 키 공간을 분할한 일부를 키 블록이라 하는데 이것은 여러 개의 키를 묶어 표현한 것으로써 예를 들면, 10비트 키 길이에 대해서 64(2^6)개의 키를 하나로 묶어서 16(2^4)개의 키 블록으로 분할할 수 있다. 이와 같이 다수의 클라이언트가 할당받은 각각의 키 블록을 동시에 조사함으로써 전수 조사에 소요되는 시간을 크게 줄일 수 있다.

키 공간 탐색을 위한 클라이언트/서버 시스템을 설계하기 위해 외국의 구현 사례를 분석하여 키 탐색 시스템에서 고려해야 할 사항을 정립하고, 이를 기반으로 키 탐색에 있어서 매우 중요한 요소인 키 블록 할당 알고리즘의 설계를 통해 구현한 클라이언트/서버 시스템을 살펴보면 다음과 같다.

3.1 구현 사례

(1) Distributed.net 팀

미국의 Distributed.net팀은 RC5 56비트 키 찾기를 찾아낸 팀인데, 현재 RC5 64비트 키 찾기를 수행 중이다. 이 팀에서 운영하고 있는 키 탐색 시스템의 구성도는 (그림 4)와 같다.



(그림 4) Distributed.net 시스템 구성도

키 할당 서버는 클라이언트에게 키 블록을 나누어 주는 역할을 하는데, 여러 개의 키 할당 서버를 운영할 수 있는 시스템으로 구현하였으며, 현재 RC5/64비트 암호키 찾기를 위해 미국, 유럽, 아시아 등의 네트워크 도메인에서 7개 정도의 키 할당 서버를 운영하고 있다. 이와 같이 구현함으로써 키 블록의 할당에 관한 정보는 서버 간의 통신으로 정보를 교환하고 있다. 이 때 클라이언트에게 할당하는 키 블록의 크기는 28~31비트 사이에서 선택할 수 있도록 하고 있다.

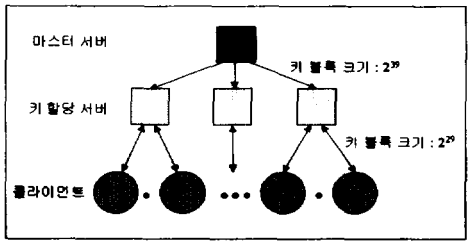
이 시스템은 하나의 서버가 다운될 경우에 다른 서버를 통해 계속적으로 키 블록을 할당받을 수 있다는 장점이 있지만 클라이언트에게 키 블록을 할당해 주기 위해서 서버 사이의 통신을 통해 키 블록 할당 정보를 서로 교환해야 하기 때문에 서버의 구조 및 프로토콜이 복잡해지고, 현재 진행 중인 키 공간에 대한 키 탐색 진행 속도, 키 할당 현황 등과 같은 정보를 클라이언트에게 제공하기 어렵다.

(2) SolNET DES 팀

스웨덴의 SolNET DES 팀에서 개발한 SolNET DES 시스템은 DES 56비트 키를 찾기를 수행하는 도중에 다른 팀에서 먼저 키를 찾아 중단된 시스템이다. 이 팀에서 운영한 시스템은 (그림 5)와 같다.

이 시스템은 Distributed.net시스템 구성과는 크게 다른 시스템으로써 키 할당을 위해서 마스터 서버와 키 할당 서버를 독립적으로 운영하고 있다. 즉 키 할당 여부에 관련된 정보를 관리하기 쉽도록 하나의 마스터 서버를 두고, 하위의 키 할당 서버를 통해 클라이언트에게 키 블록을 나누어 주는 형태이다. 이 시스템의 경우는 마스터 서버를 두어 키 블록 관리가 쉽다는 장점이 있지만, 마스터 서버가 다운될 경우에 키 할당 서버가 모두 중단될 수 있는 큰 단점이 있다. 그러나 키 할당 서버의 구조 및 프로토콜이 Distributed.net시스템에 비하여 단순한 시스템이다.

이 시스템의 경우 클라이언트에게 키 블록을 할당하기 위해서는 키 할당 서버는 먼저 마스터 서버로부터 39비트 길이의 키 블록을 받아서 이것을 29비트 크기로 다시 분할하여 키 블록을 할당하는 방식을 이용하고 있다. 이와 같은 할당 방식으로 인해 시스템을 구성하고, 운영하는 측면에 볼 때 복잡하다. 그러나 이처럼 중앙 집중식으로 키 블록 할당 정보를 관리할 경우 Distributed.net시스템과는 달리 현재 진행 중인 키 공간에 대한 키 탐색 진행 속도, 키 할당 현황 등과 같은 정보를 클라이언트에게 제공하기 쉬워진다.



(그림 5) SolNET 시스템 구성도

3.2 설계시 고려 사항

외국의 시스템을 분석해 본 결과 키 공간 탐색을 위한 클라이언트/서버 시스템은 다음과 같은 조건을 충족해야 한다.

3.2.1 서버 시스템

가) 키 공간에서 추출한 키들을 적절히 선택하여 할당하는 것은 효율성 측면에서 중요한 요소로 작용하기 때문에 키 공간을 적절한 크기로 가변 분할하여 할당할 수 있어야 한다. 즉, 클라이언트의 컴퓨팅 환경에 적절한 키 공간을 할당할 수 있어야 한다.

나) 이미 클라이언트에게 할당한 키에 대해서 결과를 받지 못할 경우 재할당이 가능해야 한다. 그렇지 못할 경우 조사되지 않은 키가 발생할 수 있다.

다) 네트워크 전송 오류에 대비하여 키 블록 및 처리 결과를 전송할 때 무결성을 보장해야 한다.
라) 현재 키 공간 탐색에 따른 진행 결과 및 수행시간을 사용자가 볼 수 있도록 웹 서비스를 통해 제공한다.
마) 서버 시스템이 다운될 경우 키 탐색 작업이 영향을 받지 않도록 클라이언트는 서버와 오프라인(Off-line) 상태에서 키 탐색을 수행하도록 해야 한다.

3.2.2 클라이언트 시스템

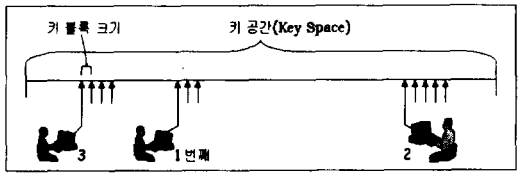
가) 클라이언트가 복호화한 메시지가 의미있는 메시지인지 확인할 수 있어야 한다. 즉, 암호화에 사용된 키를 찾았는지에 대한 확인 작업이 효율적으로 이루어져야 한다. HTML 문서, 또는 전자우편 등과 같은 아스키(ASCII) 코드 형태의 메시지 뿐 만 아니라 다양한 문서 형식를 식별할 수 있도록 고려해야 한다.
나) 수행 도중 사용자의 강제적인 중단, 또는 다운될 경우를 대비하여 키 탐색 결과를 적절히 기록하고 있어야 한다.
다) 수행속도를 고려하여 키 블록의 크기를 가변적으로 선택하여 받을 수 있도록 해야 한다.
라) 사용자의 작업을 크게 방해하지 않는 범위 내에서 키 탐색을 병행할 수 있도록 CPU의 이용률을 선택할 수 있는 기능을 포함해야 하며, 사용자의 편리를 위해 가능한 모든 기능을 자동화할 필요가 있다.
마) 다수의 클라이언트가 참여할 수 있도록 다양한 운영체제에서 동작할 수 있어야 한다.

3.3 키 블록 할당 알고리즘

키 블록의 생성과 할당은 40~256 비트 사이의 다양한 키 길이를 탐색하기 위한 매우 중요한 요소인데, 키 블록의 크기를 고정 길이와 가변 길이로 설정할 수 있도록 본 논문에서는 "임의 순차 키 블록 할당 알고리즘"과 "범위지정 키 블록 할당 알고리즘"의 두 가지 방식을 설계, 제안하였다. 따라서, 40~256 비트 사이의 다양한 키 길이를 전수 조사하는 것이 가능하다. 키 탐색은 키 공간을 모두 조사하는 것보다는 가능한 한 빨리 암호키를 찾는데 목적이 있기 때문에 이처럼 두 가지의 키 블록 할당 알고리즘을 적용해 봄으로써 키 공간 내에서 임의의 위치에 존재하는 암호키를 효과적으로 탐색할 수 있는 방안을 고려해 보았다.

3.3.1 임의 순차 키 블록 할당 알고리즘

키 공간 내의 키 블록을 고정 길이로 정하고, 클라이언트가 처음으로 접속한 경우에는 임의의 키 블록을 할당한다. 그 다음부터는 이전에 할당된 키 블록을 기준으로 순차적인 키 블록 할당을 시도한다. 그렇지만 순차적으로 키 블록을 할당할 수 없는 경우에는 임의로 선택한 키 블록을 할당하게 한다((그림 6) 참조).



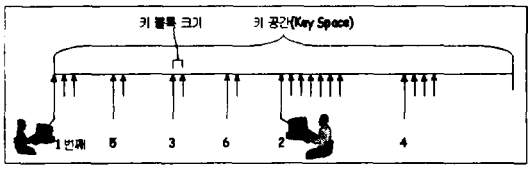
(그림 6) 임의 순차 키 블록 할당

이 방식의 경우, 키 블록의 크기를 고정 길이로 사용함으로써 여러 개의 키 서버에 분산하여 키 블록 할당 정보를 관리하는 것이 가능하며, 구현이 용이하다. 그러나 키 길이가 커짐에 따라 관리해야 할 키 블록 정보가 증가하는 단점이 있기 때문에 56비트 이하의 키 길이에 대해 적합한 방식이다. 또한 키 블록 할당 서버가 다운되었을 때에는 클라이언트에게 더 이상 키 블록을 할당하지 못함으로 이에 대한 대비책이 별도로 필요하다. 이러한 문제를 해결하기 위해 이후에 할당받을 키 블록을 미리 예약하여 할당하는 기법으로 구현하였다.

3.3.2 범위 지정 키 블록 할당 알고리즘

이 알고리즘은 키 블록의 크기를 고정하지 않고, 할당하지 않은 키 공간을 1/2씩 분할하여 키 블록의 시작과 끝 위치 정보를 이용해 키 탐색 범위를 클라이언트에게 할당하는 방식이다. 첫 번째 클라이언트가 키 할당을 요구하면 키 탐색 범위로 키 공간의 처음 위치와 마지막 위치를 알려준다. 두 번째 클라이언트는 키 공간의 1/2 위치와 마지막 위치를 알려주게 된다. 또한 새로운 클라이언트에게 범위를 지정해 줄 때 키 탐색 속도가 느린 키 공간에 우선 배치하는 방식을 적용한다. 이러한 식으로 키 탐색 범위를 할당할 경우 연결된 클라이언트 수가 증가함에 따라 각 클라이언트의 키 탐색 범위는 감소하게 되고, 키 탐색 속도가 느린 키 공간에 다수의 클라이언트들이 배치되게 된다. 여기에서 키 블록의 크기는 클라이언트가 키 탐색 결과를 전송하는 단위를 의미하는데, 키 탐색 결과를 전송할 때 서버는 클라이언트 간에

중복되는 키 탐색 범위가 있을 경우 클라이언트의 키 탐색 범위를 재지정해 주게 된다 참조((그림 7) 참조).



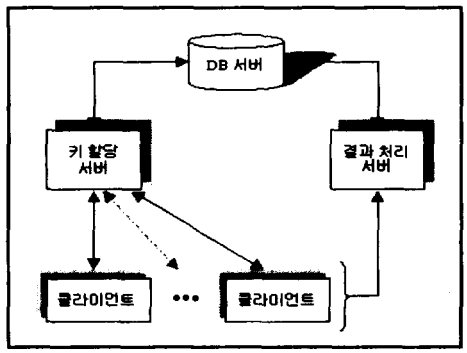
(그림 7) 범위 지정 키 블록 할당

이 방식은 가변 길이 키 블록 할당 방식을 이용함으로써 임의 순차 키 블록 할당 알고리즘과는 달리 구현이 어렵고, 키 블록 할당 정보 관리에 어려운 점이 있다. 그러나 다양한 키 길이에 적용하기 용이하며, 키 블록 할당 서버가 다운되었을 경우에도 클라이언트는 계속적으로 키를 탐색하는 것이 가능하다. 또한 클라이언트의 키 탐색 속도에 따라 적절히 키 탐색 범위를 지정하여 속도가 빠른 클라이언트에게 서버의 간섭없이 더 많은 키 탐색을 수행시킬 수 있는 장점이 있다.

즉, 클라이언트의 성능을 최대한 활용하면서 키 공간내에 균등하게 배치할 수 있는 점이 임의 순차 키 블록 할당 알고리즘과 크게 다른점이다.

3.4 시스템 구성

키 탐색에 참여하는 클라이언트들이 동시에 서로 다른 키를 탐색할 수 있도록 클라이언트/서버 시스템을 구현하기 위해서 본 논문에서는 (그림 8)에서 보는 바와 같이 크게 DB 서버, 키 할당 서버, 결과 처리 서버, 클라이언트 등 네 부분으로 구성하였다. 먼저 각 구성요소의 역할을 살펴보면 다음과 같다.



(그림 8) 키 공간 탐색을 위한 분산시스템 구성도

3.4.1 DB 서버

키 할당 서버가 클라이언트에게 키 블록을 할당하는데 필요한 정보를 관리한다. 또한 클라이언트가 전송하는 각 키 블록들의 처리 결과를 저장하고, 이를 통해 키 탐색에 관련하여 생성되는 통계 정보를 저장 및 관리한다. DB 서버는 하나만 유지되며, 하나 이상의 키 할당 서버, 결과 처리 서버 등이 독립적으로 운영될 경우 이 시스템들의 서로 간의 공유정보를 쉽게 관리하고, 통계정보를 생성하는데 필요한 정보를 효율적으로 유지하기 위해서 DB서버가 존재한다.

3.4.2 키 할당 서버

키 할당 서버는 키 블록을 다수의 클라이언트에게 나누어주는 역할을 하는데, RC4 40 비트와 같이 작은 키 길이에 적합한 키 블록 할당 방식인 "임의 순차 키 블록 할당 알고리즘"과 40~256 비트 사이의 다양한 키 길이를 전수 조사하는 것이 가능한 "범위 지정 키 블록 할당 알고리즘"을 구현하였다. 또한 "임의 순차 키 블록 할당 알고리즘" 방식의 단점인 키 할당 서버가 다운될 경우 키 블록을 할당받지 못해 키 탐색을 중단하는 문제를 해결하기 위한 방편으로 클라이언트에게 미리 다음에 탐색할 키 블록인 예비 키 블록을 함께 할당해 줄 수 있도록 설계하였다. 이렇게 함으로써 서버가 다운될 경우에 클라이언트는 할당받은 키 블록을 모두 탐색한 후 서버로부터 키 할당을 받지 못해 키 탐색이 중단되는 일 없이 미리 할당 받은 예비 키 블록을 자동으로 계속 탐색하는 것이 가능하다. 그리고 미리 할당받은 예비 키 블록 마저 키 탐색이 모두 완료된 상황에서 서버가 다운되어 있다면 클라이언트는 자동으로 서버에 연결을 재시도할 수 있도록 구현하였다.

키 탐색에 있어서 키 공간을 모두 조사하는 것이 목적이 아니라 가능한 빨리 암호키를 찾는데 있기 때문에 이와 같이 서로 다른 두 가지 할당 알고리즘을 적용함으로써 다양한 키 길이에 대해 효과적으로 키 할당이 가능하게 하여 키 공간 내에서 임의의 위치에 존재하는 암호키를 효과적으로 탐색할 수 있도록 하였다. 또한 키 블록을 할당하는데 필요한 정보는 파일 시스템을 이용할 수도 있으나, 효율적인 관리를 위해 DB 서버를 도입하여 관리하는 방식을 적용하였다.

3.4.3 결과 처리 서버(Report Processing Server)

키 블록을 할당받은 클라이언트가 키를 모두 조사한

후 처리 결과를 전송하면, 이를 DB 서버에 기록하고, 클라이언트가 키를 찾았을 경우 이것에 대한 유효성을 조사한다. 또한 현재 운영에 참여한 클라이언트의 수, 조사가 끝난 키 블록 정보, 전체 키 탐색 시간 등과 같은 통계정보를 생성하여 DB 서버에 저장하고, 통계정보를 사용자들에게 웹(WWW)을 통해 서비스해 주는 역할도 한다.

결과 처리 서버와 키 할당 서버는 하나의 시스템으로 통합하여 구현하는 것이 가능하지만, 키 탐색 시스템의 성격상 수 백 수 천대 이상의 클라이언트가 네트워크로 연결될 수 있다는 점을 고려해야 한다. 따라서 다수의 클라이언트가 키 탐색에 참여할 경우 동시에 키 할당과 결과 처리를 위해 연결되어 발생할 수 있는 병목현상을 줄이기 위해 키 할당 서버와 결과처리 서버를 분리하여 서로 다른 시스템에서 독립적으로 운영할 수 있도록 분리하여 구현하였다.

3.4.4 클라이언트

키 탐색의 실질적인 주체로써 키 할당 서버로부터 키 블록을 할당받으면 각각의 키에 대해 순차적으로 복호화를 시도한다. 처음 시도하게 될 40비트 RC4를 이용한 암호문은 HTTP 메시지를 식별하는 형태로 동작하며, 복호화를 시도하는 메시지의 형태에 따라 의미있는 복호화 메시지를 식별하는 방법이 달라질 수 있다.

그리고 모든 키를 조사한 후 수행 결과를 생성하여 결과 처리 서버에 전송한다. 만약 키 할당 서버가 다운되었을 경우 효율적으로 동작하도록 자동으로 예비 키 블록을 계속해서 탐색한다.

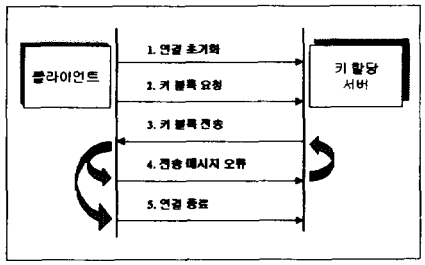
클라이언트의 수는 가능한 많을수록 모든 키 공간을 탐색하는데 소요되는 시간을 줄일 수 있기 때문에 클라이언트 프로그램은 가능한 Windows 95/NT, Linux, SUN Solaris 등과 같은 다양한 운영체제에서 수행될 수 있도록 구현하였다.

3.5 시스템 동작 시나리오

각 시스템 구성요소 사이의 동작을 설명하기 위해 키 할당 서버와 클라이언트의 동작, 결과 처리 서버와 클라이언트의 동작으로 구분하여 기본적인 프로토콜과 메시지 구조를 설명한다.

3.5.1 키 할당 서버와 클라이언트

키 할당 서버로부터 클라이언트가 키 블록을 할당받는 기본적인 프로토콜은 (그림 9)와 같다.



(그림 9) 키 할당 프로토콜

가) 연결 초기화

클라이언트가 키 할당 서버에 TCP/IP 소켓(Socket)을 이용하여 연결한다. 이 단계에서 키 할당 서버는 클라이언트 IP주소 등의 정보를 기록한다.

나) 키 블록 요청

클라이언트가 키 할당 서버에 키 블록을 요구한다.

Mode	암호 알고리즘 식별자
Request Block SIZE	전송받을 키 블록의 크기(키의 개수)
Reserved Block#	전에 예약된 키 블록 식별자, 초기값 : 0
User eMail	사용자 전자우편 주소

다) 키 블록 전송

키 할당 서버는 클라이언트에게 키 블록 할당 알고리즘에 따라 키 블록을 선택하여 아래의 메시지를 클라이언트에게 전송한다. 여기에서 "Authenticate" 필드의 Nonce값은 처리 결과 메시지를 생성할 때 이용되는 중요한 정보인데, 이 값은 키 블록마다 하나씩 유일하게 부여된다.

Mode	암호 알고리즘 식별자
Key Size	암호 알고리즘에서 사용하는 키의 길이
Key Block#	키 블록 식별자
Key Block	키 블록; 예) 0xfa 0x34 0x00 0x00 0x00
Reserved Block#	예약된 키 블록 식별자
Reserved Block	키 블록; 예) 0xfa 0x35 0x00 0x00 0x00
Encrypted Message	복호화에 사용할 암호문
Message Certificate	Hash(Key Block, Reserved Block#, Reserved Block)
Block Assign Server	다른 키 할당 서버의 IP 주소 (추가적으로 운영될 경우)
Authenticate	Nonce(처리 결과 전송시 필요)
Report Processing Server	결과 처리 서버의 IP 주소

라) 전송 메시지 오류

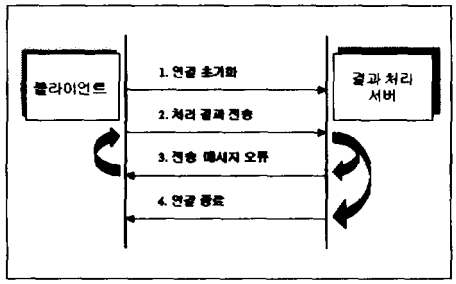
키 할당 서버로부터 할당받은 메시지 내의 "Message Certificate" 값을 이용해 데이터의 변조 여부를 검사한 후 오류 발생시 키 블록 재전송을 요구한다.

마) 연결 종료

TCP/IP 소켓 연결을 종료하고, 키 조사를 시작한다. 클라이언트는 키 할당 서버가 다운되었을 경우 예약된 키 블록이 있으면 이를 계속 탐색한다.

3.5.2 결과 처리 서버와 클라이언트

클라이언트는 할당받은 키 블록을 모두 조사한 후에 (그림 10)과 같은 프로토콜을 이용해서 처리 결과를 결과 처리 서버에 전송한다.



(그림 10) 결과 처리 프로토콜

가) 연결 초기화

클라이언트는 1)의 다) 메시지에 포함된 결과 처리 서버의 IP주소를 참조하여 연결한다.

나) 처리 결과 전송

클라이언트는 할당받은 키 블록을 모두 조사한 후 결과를 전송한다. 이 메시지 중 "Report Certificate"의 값은 1)의 다) 메시지에 포함된 "Authenticate" 정보

Mode	암호 알고리즘 식별자
Report Type	암호키 찾기의 실패 여부
Key Block#	조사한 키 블록
Key	찾은 암호키, 예) 0xfa 0x34 0x01 0xa7 0xc4
Decrypted Message	암호키로 복호화한 평문
Execution Time	키 조사에 걸린 수행 시간
Report Certificate	Hash(Report Type, Key Block#, Key, Decrypted Message, 1)의 다) "Authenticate")

(Nonce값)를 알지 못하면 생성할 수 없는 정보로써 처리 결과를 인증하기 위해 사용된다. 따라서 클라이언트는 할당받은 키 블록에 대해서만 그 처리 결과를 생성할 수 있도록 하였다. 이렇게 함으로써 자신이 할당받지 않은 키 블록에 대해 키 탐색을 수행한 것처럼 임의의 처리 결과를 생성하여 전송하는 것은 불가능하다.

다) 전송 메시지 오류

클라이언트가 전송한 "Report Certificate" 값을 검사하여 무결성을 검사한 후 오류 발생시 재전송을 요구한다. 클라이언트가 할당받은 키 블록에 대해서 처리 결과를 정확하게 전송하지 않는다면 키 공간을 모두 조사하고도 키를 찾지 못하는 심각한 문제가 발생할 수 있다. 이와 같은 상황에서 임의의 클라이언트가 자신이 키 탐색을 하지 않고 한 것처럼 속이거나 오동작에 의해 특정 키 블록에 대한 처리 결과를 만들어 전송할 경우 오류가 발생하고, 그 처리 결과는 무시된다.

라) 연결 종료

TCP/IP 소켓 연결을 종료하고, 자동으로 키 할당 서버에 연결하여 키 블록을 전송 받는다. 이때 클라이언트는 키 할당 서버가 다운되었을 경우 예약된 키 블록을 계속적으로 탐색한다.

4. 시스템 구현

먼저 키 공간을 전수 조사하는 클라이언트/서버 시스템을 구현하기 위한 개발 환경을 살펴보면 <표 4>와 같다.

〈표 4〉 클라이언트/서버 시스템 개발 환경

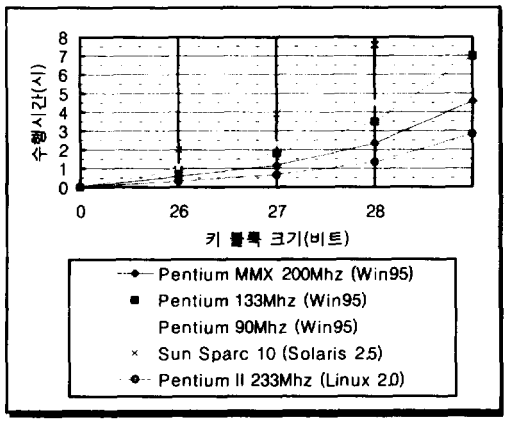
구 분	하드웨어	소프트웨어
DB서버	Sun Sparc 10	운영체제 : SUN Solaris 2.5 데이터베이스 : MySQL 3.21
키 할당 서버 결과 처리 서버	Sun Sparc 10	운영체제 : SUN Solaris 2.5 컴파일러 : Gcc 2.7.2
클라이언트	Pentium 133㎒	운영체제 : Windows 95 컴파일러 : Borland C++ Builder

이와 같은 환경에서 구현한 키 공간 탐색을 위한 클라이언트/서버 시스템을 운영하기에 앞서 키 블록에 대한 크기 범위를 미리 결정할 필요가 있다. 이렇게

하여 클라이언트 사양에 적절한 키 블록 크기를 제공함으로써 효과적으로 클라이언트 시스템을 이용할 수 있기 때문이다.

(그림 11)은 키 블록의 길이에 대한 범위를 결정하기 위한 데이터로써 본 논문을 통해 구현한 클라이언트를 이용해 26, 27, 28비트 길이의 키 블록을 모두 조사하는데 시스템 사양에 따라 걸린 시간을 측정한 것이다. RC4 암호 알고리즘을 이용해 72바이트의 평문을 암호화하여 복호화를 위한 암호문을 생성하였으며, 이 결과는 시스템의 CPU를 가능한 최대한 사용해서 얻은 것인데 실제는 그 이상의 시간이 걸릴 것으로 추측된다.

SUN Sparc 10 시스템의 경우에는 수행 시간이 7시간 이상 소요되었는데 이 시스템은 서버 데몬(Daemon) 이 다수 동작하고 있어서 비교적 많은 시간이 소요되었다. 이와 같이 비연속적으로 시간을 할애하여 키 탐색을 할 경우 시스템 사양에 따라 1일 이상이 걸리는 것도 있을 것으로 예측된다.

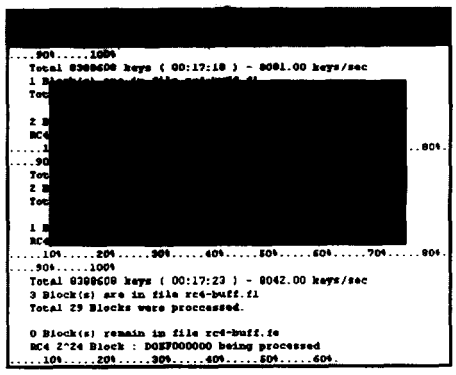


(그림 11) 키 블록 길이에 따른 수행 시간

그리고 암호키 찾는데 소요되는 시간은 참여하는 클라이언트의 수에 따라 많은 영향을 받지만 RC4 40비트 암호키는 Pentium II 233Mhz를 사양으로 하는 클라이언트 4096대가 동시에 28비트 길이의 키 블록을 조사할 경우 1시간 30분 내, Pentium MMX 200Mhz의 경우는 3시간 내에 암호키를 찾을 수 있음을 예측할 수 있다. 또한 프로세서를 항상 100% 이용할 수 있다는 가정하에서 i비트 길이의 키 블록을 모두 조사하는데 걸리는 시간(C_i)을 계산할 수 있으면, n비트 길이의 키 블록을 모두 조사하는데 소요되는 시간(C_n)은 계산식 $C_n = C_i * 2^{n-i}$으로 예측할 수 있다. 이 식을 적용하여 Pentium II 233Mhz 프로세서가 28비트를 조사하는데 소요된 1.3시간을 대입하여 32비트 길이의 키 블록을 모두 조사하는데 소요되는 시간을 계산해 보면 20.8($1.3 * 2^{32-28}$) 시간이 소요됨을 추정할 수 있다.

본 논문에서 구현한 키 탐색 클라이언트 (그림 12)는 RC4 알고리즘의 40비트 키 길이에 대한 키 블록 길이의 범위는 클라이언트의 성능을 고려하여 26~28비트 사이에서 선택할 수 있도록 하였다.

56~256비트 이상의 키 길이에 대해서는 클라이언트에게 할당하는 키 블록의 크기를 보다 크게하여 키 블록의 수를 적정 수준으로 줄임으로써 키 할당 서버에 접속하는 빈도를 줄이는 것이 효과적이다. 따라서 키 블록 길이의 범위는 28~32비트 사이에서 결정할 수 있도록 하였다.



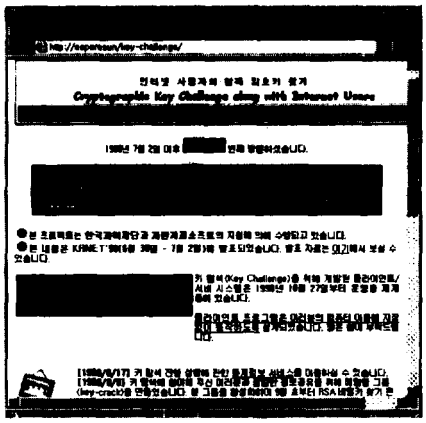
(그림 12) 키 탐색 클라이언트 어플리케이션

5. 실험 결과

본 논문에서 구현한 키 찾기 클라이언트/서버 시스템을 이용하여 RC4 알고리즘의 40비트 키 찾기를 수행하였다. 이를 위해 먼저 (그림 13)과 같이 "암호키 찾기 홈페이지"를 구축하여 인터넷 사용자들에게 암호키 찾기에 참여하도록 유도하였다.

암호키 찾기에 사용한 암호문은 40비트 길이의 임의의 키를 생성한 후 RC4 알고리즘을 이용하여 <표 5>와 같이 생성하였다.

이와 같은 암호문에 사용된 키를 찾기 위해 3.다에서 제시한 키 블록 할당 알고리즘을 이용하여 클라이언트에게 키 블록을 할당하여 수행하였다.



(그림 13) 암호키 찾기 홈페이지

〈표 5〉 암호문(16진수)

45 6F DF F0 15 25 6D 5D 7B 7E 38 0E 12 A2 EE A7 EB 39 CE 69 EE 0A 3C EB DA 7E 87 8B C6 25 4D E0 99 59 25 24 37 3D E4 E9 DD 53 C2 06 D8 B9 B3 27 F9 82 75 A6 83 C5 A5 83 18 84 F7 72 88 57 F5 9F 54 9C C7 B1 2C A6 D3 2B 7F

5.1 암호키 찾는데 소요된 시간

암호키 찾기 시작 : 1998년 10월 27일 16시 29분 26초
암호키 찾은 시간 : 1998년 11월 03일 23시 21분 45초

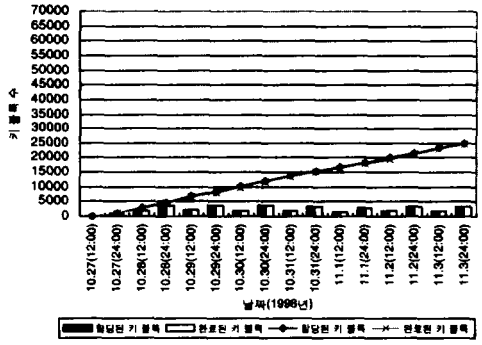
총 61대의 클라이언트가 참여하여 이루어진 암호키 찾기의 전체 수행 시간은 174시간 52분 19초 였는데, 이것을 날짜로 환산하면 약 7.5일 만에 암호키를 찾은 것이다.

인터넷 주소가 dynasty.chungnam.ac.kr인 Pentium MMX 200Mhz 컴퓨터에서 암호키를 찾는 행운을 얻었는데, 찾아낸 암호키는 "a9 4d 4d 68 08"(16진수 표현), 알아낸 평문 "unknown message is : Strong cryptography makes the world a safer place."이었다.

5.2 키 탐색 결과

키 블록의 크기는 2^24으로 고정하여 40비트 키 길이에 대하여 2^16개의 키 블록을 생성하여 사용자에게 임의 순차 키 블록 할당 알고리즘에 의해 클라이언트에게 키 블록을 할당해 주었는데, 키 공간 2^16 중 39% 정도를 조사하여 키를 찾아냈다. 다음 그림은 날짜별 키 블록 할당 개수와 조사를 완료한 키 블록의 개수를 나타내고 있다((그림 14) 참조).

암호키 찾기 수행 동안 조사를 완료한 키의 개수는 4,1830,6326,528(24,933 * 2^24)인데, 이러한 수행 속도는 초 당 약 664,465개(24,933 * 2^24/174시간 52분 19초)의 키를 조사한 속도이다. 본 시스템에서 40비트 키 길이에 대해 사용한 임의 순차 키 블록 할당 알고리즘은 키 공간의 39% 정도만 탐색한 후 키를 찾아냈는데, 평균적으로 키 공간의 50% 이상을 조사 한 후 키를 찾아 낼 수 있을 것으로 기대되는 상황에서 이것은 상당히 운이 좋은 경우에 속한다.



(그림 14) 키 블록 할당 및 탐색

6. 결 론

현재 이용되고 있는 대부분의 암호 알고리즘은 공개되어 있기 때문에 암호 알고리즘에 대한 안전성은 암호화를 수행할 때 사용된 키의 길이에 의존하게 된다. 즉 키 공간을 전수조사를 통한 공격이 가능하기 때문에 안전한 길이의 키를 사용해야 한다. 이러한 전수조사를 통한 공격 방식은 초기에 하나의 컴퓨터를 이용해 구현하거나, H/W를 이용한 전수조사 전용 시스템 구현을 통해 시도되었는데, 이러한 전수조사 방법은 시스템 개발비용이 너무 높을 뿐 만 아니라 처리 속도의 문제점이 있다.

그러나 하드웨어 수행속도와 분산처리 기술이 발전함에 따라 여러 대의 컴퓨터를 네트워크로 연결하여 암호 알고리즘에서 사용하는 키 공간을 각각의 클라이언트 시스템에서 나누어 조사해 봄으로써 암호문을 만드는데 사용된 키와 평문을 좀 더 빠르게 찾아 내는 것이 가능하게 되었다. 즉 전수조사를 위해 개발된 전용 시스템에서 키를 탐색하는 것이 아니라 여러 대의 시스템에서 분산 처리하여 키 탐색에 걸리는 시간을 줄일 수 있게 된 것이다.

이와 같은 시도는 미국을 중심으로1997년 1월에 시작한 "RSA Secret-Key Challenge Contest"를 계기로 본격적으로 이루어지게 되었다. 현재 키 전수 조사 시스템을 개발하고 있는 대표적인 그룹은 크게 Distributed.net, Electronic Frontier Foundation(EFF), SolNet 등이 있는데, 자신들이 독자적으로 설계하여 개발한 시스템에 대한 소스코드를 공개하거나, 국외유출을 제한하고 있는 상황이다. 따라서 이러한 시스템에 대한 구체적인 내용을 확인하는 것이 불가능하며, 단지 키 전수조사에 분산처리 기술을 도입했다는 것과 전체 시스템 구성에 대한 정보만을 확인할 수 있다[4].

본 논문에서는 키 전수조사를 위해 전용시스템 개발이 아닌 분산처리 기술을 도입하여 현재 메시지의 암호화에 주로 이용되고 있는 관용 암호 알고리즘에서 사용되고 있는 40~256 비트 사이의 다양한 키 길이에 대해서 키 공간 전체를 조사하여 암호화에 사용된 키를 찾는 클라이언트/서버 시스템을 구현하였다.

이처럼 다양한 키 길이에 대한 전수 조사를 효과적으로 수행하기 위해 "임의 순차 키 블록 할당 알고리즘"과 "범위 지정 키 블록 할당 알고리즘"을 독자적으로 설계하였다. 키 전수 조사 시스템은 키 공간 전체를 모두 탐색하는 것이 아니라 가능한 빨리 키를 찾아내는 것이기 목적이기 때문에 클라이언트에게 어떠한 순서로 키 블록을 할당하는 것이 보다 효율적인지 고려해서 개발되어야 한다. 또한 이러한 알고리즘은128비트 길이와 같이 매우 큰 키 길이에 대해서도 키 할당이 가능해야 한다.

즉 2^{128}(340,282,366,920,938,463,463,374,607,431,770,000,000)개의 키에 대한 키 블록을 생성하고, 이에 대한 키 탐색 정보를 관리하여 클라이언트에게 키 할당을 할 수 있어야 하는 문제는 매우 중요하다.

이를 위해 본 논문에서는 상대적으로 작은 56비트 이하의 키 길이에 대한 전수조사를 수행하는데 적합한 "임의 순차 키 블록 할당 알고리즘"을 개발하고, 64비트 이상의 큰 키 길이에 대한 전수조사에 효율적으로 동작할 수 있는 "범위 지정 키 블록 할당 알고리즘"을 개발하였다.

이와 같은 키 할당 알고리즘을 구현한 서버와 클라이언트는 분산처리 기술을 이용해 다수의 클라이언트가 키 탐색에 동원되어 키 탐색을 빠르게 처리할 수 있으며, 각 클라이언트는 가능한 사용자 작업에 영향을 주지 않는 범위에서 즉, CPU를 사용하지 않는 시간에 키 탐색을 할 수 있도록 개발하였다.

이 시스템으로 시도한 RC4/40비트 키 길이에 대한 암호키 찾기에서는 임의 순차 키 블록 할당 알고리즘을 이용해 클라이언트에게 26~28비트 길이의 키 블록을 할당하여 키 찾기를 수행한 결과 키 공간의 39.9%를 조사한 후 약 7.5일 만에 암호키를 찾아내는 성과를 거두었다.

향후 RC4/40비트 암호키 찾기를 성공적으로 마친 암호키 찾기 시스템을 이용하여 RSA사에서 현재 진행중인 암호키 찾기 콘테스트에 참여할 계획이다.

이와 같은 시도를 통해 국내에서 널리 이용되고 있는 SSL기반의 외국 암호 시스템의 위험성을 보임으로써 국내의 전자우편, 전자상거래 환경 등에서 안전한 암호시스템을 이용하도록 유도하는 계기가 될 것으로 기대된다.

참 고 문 헌

[1] Bruce Schneier, *Applied Cryptography Second Edition*, John Wiley & Sons, 1996, pp.265-356.

[2] William Stallings, *Network and Internetwork Security Principles and Practice*, Prentice-Hall, 1995, pp.21-156.

[3] MattBlaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener, Minimal Key Lenths for Symmetric Ciphers to Provide Adequate Commercial Security, A Report by an Ad Hoc Group of Cryptographers and Computer Scientists, 1996, http://theory.lcs.mit.edu/~rivest/bsa-final-report.ps.

[4] The RSA Data Security Secret-Key Challenge, http://www.rsa.com/rsalabs/97challenge/, RSA Corp..

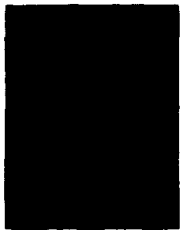
[5] http://adams.patriot.net/~johnson/html/neil/sec/cryptanalysis.htm.

[6] http://www.cl.cam.ac.uk/brute/challenge/.

[7] The SSL Protocol Version 3.0 Internet Draft, IETF.

[8] SSL3.0 Implementation Assistance, http://home.

netscape.com/eng/ssl3/traces/index.html, Netscape Corp..
[9] SSL Challenge, http://pauillac.inria.fr/~doligez/ssl/.



송 상 헌

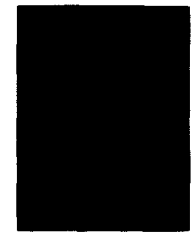
e-mail : shsong@esperosun.chungnam.ac.kr
1996년 배재대학교 전자계산학과 졸업(학사)
1998년 충남대학교 대학원 컴퓨터과학과(석사)
1998년~현재 충남대학교 대학원 컴퓨터과학과 박사과정
관심분야 : 보안 프로토콜, PKI, 전자지불시스템

박 현 동

e-mail : hdpark@esperosun.chungnam.ac.kr
1995년 충남대학교 전산학과(학사)
1997년 충남대학교 대학원 컴퓨터과학과(석사)
1997년~현재 충남대학교 대학원 컴퓨터과학과 박사과정
관심분야 : 네트워크 보안, 암호학

김 충 길

e-mail : chkim@esperosun.chungnam.ac.kr
1998년 충남대학교 물리학과(학사)
1998년~현재 충남대학교 대학원 컴퓨터과학과 석사과정
관심분야 : 운영체제 보안, 분산처리



박 중 길

e-mail : jgpark@sunam.kreonet.re.kr
1986년 동국대학교 전자계산학과(학사)
1988년 서강대학교 전자계산학과(석사)
1988년~현재 국방과학연구소 선임연구원
1998년~현재 충남대학교 대학원 컴퓨터과학과 박사과정
관심분야 : 컴퓨터통신보안, 접근통제, 암호이론

김 영 진

e-mail : knight@sunam.kreonet.re.kr
1989년 중앙대학교 전자계산학과(학사)
1998년~현재 충남대학교 컴퓨터과학과 석사과정
관심분야 : 컴퓨터통신보안, 스마트카드 응용, 정보보호 이론

류 재 철

e-mail : jcryou@esperosun.chungnam.ac.kr
1985년 한양대학교 산업공학과(학사)
1988년 Iowa State University 전산학과(석사)
1990년 Northwestern University 전산학과(박사)
1991년~현재 충남대학교 컴퓨터과학과 부교수
관심분야 : 컴퓨터통신보안, 전자상거래, 분산처리