인터넷을 이용한 분산 처리와 정보보호

이성룡° 주한규
한림대학교 정보통신공학부
{srlee, hkjoo}@hallym.ac.kr

Distributed Computing Using Internet and Information Security

Sungryong Lee° Hankyu Joo
Div. of Information and Communication, Hallym University

요 약

인터넷의 사용이 증가함에 따라 인터넷을 이용한 전자 상거래, 인터넷 뱅킹 등의 응용 분야가 확장되고 있다. 또한 인터넷에 연결된 많은 컴퓨터들이 분산 처리를 위해 사용될 수도 있다. 전자 상거래에서와 마찬가지로 분산 처리에서도 정보보호의 개념은 매우 중요하다. 인터넷은 누구나 접근 가능하므로 이를 통과하는 자료는 언제든지 누출될 위험이 있다. 따라서 분산 처리 시 인터넷을 통과하는 자료에 기밀성을 줄 필요가 있다. 또한 분산 처리에 사용되는 컴퓨터에 대한 인증과 자료의 무결성을 보장할 수 있는 방법이 필요하다. 인증과 무결성 또한 정보보호 기술을 사용하여 이룰 수 있다. 본 고에서는 단순한 분산처리 시스템에서의 정보보호 기술에 대하여 기술한다.

1. 서론

인터넷이라고 불리는 네트워크에 의하여 정부, 학교, 연구소, 회사, 그리고 가정의 컴퓨터들이 연결되어 있다. 이러한 인터넷 때문에 지리적으로 떨어져 있는 컴퓨터에 어려움 없이 접근할 수 있다.

여러 대의 컴퓨터를 이용하여 하나의 문제를 해결하는 분산 처리는 계산의 속도를 값싼 비용으로 증가시킬 수 있다. 암호문이 주어졌을 때, 전수조사를 이용하여 그 암호의 키를 찾는 프로그램은 하나의 컴퓨터를 이용하여 수행하는 것은 현실적으로 불가능 하다. 이러한 경우 다수의 컴퓨터를 이용하여 분산처리 함으로써, 계산 속도를 증가시킬 수 있다. 실제 암호화 키를 찾는 키 전수조사 시스템이 인터넷을 이용한 분산 처리 시스템으로 개발되었다[1, 2].

인터넷을 이용한 분산처리는 비용에 있어서의 장점과 유연성의 장점이 있다. 하지만 인터넷이라는 공공망을 이용하므로 정보 누출, 인증과 무결성의 보증 등의 문제를 해결할 수 있는 정보보호의 필요성이 대두된다.

본 고에서는 정보보호 기능을 가진 간단한 분산처리 시스템에 대하여 기술한다.

2. 관련 연구

2.1 분산 처리

분산 처리를 사용하는 하나의 예는 암호 키 탐색이다. 관용 암호 알고리즘에서 사용되는 암호 키를 찾는 기본적인 방법으로 키 전수 조사 방법이 있다. 이는 키 공간에 있는 모든 키를 이용하여 암호문을 복호화하여 봄으로써 암호 키를 찾아내는 방법이다. 이 경우 암호 키의 길이가 길어짐에 따라 수행 시간이 급속도로 길어지는 문제점이 있다. 즉 40 비트 키 길이를 가지는 경우 2^{40} 의 키 공간을 탐색하여야 하며 56 비트 키를 사용하는 경우 2^{56} 의 키 공간을 탐색하여야 한다. 이렇게 막대한 수행 시간이 필요한 경우, 수행 시간 단축을 위하여 분산 처리를 사용할 수 있다. 암호 키 탐색을 위한 대표적인 시스템은 distributed.net에 의해 개발된 키 탐색 시스템이다[1]. 국내에서도 클라이언트/서버 시스템이 발표되었다[2]. 이들 시스템은 인터넷에 연결된 다수의 컴퓨터들을 이용하여 분산처리 하는 시스템으로 평소에 각각 자신의 목적으로 사용되던 컴퓨터들이 필요 시 분산 처리를 위하여 사용될 수 있다. 이러한 분산 처리는 비용의 증가 없이 다수의 컴퓨터를 이용하여 분산 처리 함으로써 속도를 증가시킬 수 있는 장점이 있다. 하지만 인터넷을 이용한 분산 처리는 인터넷이라는 공공망을 이용하므로 정보 누출의 문제, 인증과 무결성 확인의 어려움 등의 문제가 있다.

2.2 정보 보호

정보 보호 기술을 이용하여 인터넷이라는 공공망을 사용하면서도 필요한 컴퓨터만이 연결된 사설망과 같

이 사용 할 수 있다. 암호화 기법은 자료에 기밀성을 주기 위하여 사용된다. 뿐만 아니라 암호화 기법에 사용되는 암호 알고리즘은 인증과 무결성 확인을 위하여도 사용된다. 암호화 기법은 평문 자료 m을 전송하기 전에 암호키 e를 이용하여 암호화된 자료 c로 바꾸어 준다. 암호화된 자료 c는 복호 키 d를 가진 수신자에 의하여 복호되어 평문 m이 된다. 복호 키 d를 가지지 않은 사람은 암호화된 자료 m을 획득할지라도 평문 자료 m을 얻을 수 없다. 따라서 암호화 기법에서 안전한 암호 알고리즘을 사용하고 암복호 키만 안전하게 유지되면 자료의 기밀성이 보장된다고 가정할 수 있다.

암호화 기법은 대칭키 방식과 공개키 방식의 두 종류로 나눌 수 있다. 대칭키 방식의 암호화 기법에서는 암호 키로부터 복호 키가 쉽게 구해질 수 있으며 일반적으로 암호 키와 복호 키는 동일하다. 따라서 교신하는 두 상대방이 동일한 키를 가지고 있다. DES[3], FEAL[4], IDEA[5], RC496], RC5[6], 그리고 SEED[8] 같은 암호 알고리즘은 대칭키 방식을 사용한다. 대칭 키를 사용하는 경우 키 교환과 키 관리에 어려움이 따른다. 공개키 방식의 암호화 기법에서는 암호 키로부터 복호 키를 구하는 것이 현실적으로 불가능하다. 따라서 메시지의 수신자는 복호 키를 안전하게 유지하며 송신자에게 암호 키를 이용하여 암호화하여 자료를 보낼 것을 요청한다. 암호 키로 복호 키를 알 수 없으므로 암호 키로 암호화 된 자료를 수신자 외에는 복호화 할 수 없다. 따라서 암호 키는 공개적으로 알려져도 안전하다 (공개적으로 알려져도 되는 암호 키를 공개키, 공개되어서는 안되는 복호 키를 비밀키라고 부른다). 따라서 대칭키 암호화 기법의 키 교환의 어려움이 덜어진다. 또한 관리해야 하는 키의 수가 줄어드므로 키 관리의 어려움 또한 덜어진다. 그러나 한편으로 메시지의 송신자는 사용하려는 암호 키(공개키)가 실제 메시지 수신자의 공개키인가를 확신 할 수 있는 방법을 필요로 한다. 공개키들이 특정한 사람의 실제 공개키임을 인증해 주기 위해 인증 기관(Certificate Authority - CA)이 필요하다. RSA[9], ElGamal[10] 등이 대표적인 공개키 방식의 암호화 기법이다. Diffie-Hellman 키 교환 방식[11]은 공개키 개념을 사용한 키 교환 방법이며 DSA[12]는 공개키 개념을 사용한 전자 서명 기법이다. 공개키 방식 암호화 기법은 그러나 암복호화에 너무 많은 시간을 필요로 하므로 큰 자료를 암호화 하는데 사용할 수 없다.

자료가 전송 도중 변화되지 않았는가를 확인하는 무결성과 자료가 가장된 전송자로부터 전송되지 않았음을 확인하는 인증을 위해 해쉬 함수가 사용될 수 있다. 암호학적인 해쉬 함수는 일방향 함수로 해쉬 값으로부터 원 자료를 복원하는 것은 불가능하다. 자료와 함께 그에 대응되는 해쉬 값을 보내면 수신자는 자료로부터 해쉬 값을 계산하여, 수신된 해쉬 값과 계산된 해쉬 값이 동일하면 자료가 전송 도중 변경되지 않았다고 볼 수 있다. 해쉬 함수는 키를 사용하지 않는 해쉬 함수와 키를 사용하는 해쉬 함수로 나뉜다. 키를 사용하는 해쉬 함수의 경우 동일한 자료에 대하여 언제나 동일한 해쉬 값을 생성한다. 따라서 키를 사용하지 않는 해쉬 함수 만으로는 무결성과 인증을 보장할 수 없다. 즉, 공격자가 자료를 변화 시키고 그로부터 해쉬 값을 생성할 수 있다. 키를 사용하지 않는 해쉬 함수는 암호 알고리즘과 함께 사용되어 무결성과 인증을 할 수 있다. MAC(Machine Authentication Codes) 라고도 불리는 키를 사용하는 해쉬 함수는 키를 공유한 당사자들 만이 동일한 자료로부터 동일한 해쉬 값을 생성할 수 있다. 따라서 MAC은 자료의 무결성과 인증을 보장할 수 있다. MD-5[13]와 SHA-1[14] 등이 대표적인 키를 사용하지 않는 해쉬 알고리즘이며 HMAC-MD5와 HMAC-SHA-1 등이 대표적인 MAC 알고리즘이다[15].

3. 정보보호 기능을 가지는 분산 처리

인터넷으로 연결된 컴퓨터들을 평소에는 각자의 작업을 하는데 사용하고 필요시에 분산 처리를 하는 경우 정보 보호의 필요성이 대두 된다. 네트워크에 연결된 다른 컴퓨터들에 의해 분산 처리를 위해 소통되는 자료가 누출될 수 있는 위험이 있으므로 기밀성을 줄 필요가 있다. 또한 공격자에 의해 통신 되는 자료의 불법 변경이 있을 위험이 있으므로 무결성 확인 기능이 필요하며, 공격자에 의한 불법 결과 생성을 방지하기 위하여 인증의 기능이 필요하다.

정보보호 기능을 갖춘 분산 처리 시스템이 개발 중에 있다. 특징적인 정보보호 기능은 일회용 Diffie-Hellman 키 교환 방식을 사용하며, 실제 자료 암호화를 위하여 RC4 암호 알고리즘을 사용한다. HMAC-MD5를 사용하여 무결성을 지원하며, DSA 전자 서명이 상호 인증을 위하여 사용된다. 하나의 인증(CA) 서버가 사용되어 참가하는 컴퓨터들의 공개키를 인증한다.

3.1 시스템 구성

분산 처리 시스템의 구성은 그림 1과 같다. 평소 컴퓨터들은 각각의 용도로 쓰이며, 분산 처리가 필요한 경우, 그 처리를 위해 사용될 필요한 그리고 신뢰된 컴퓨터들이 분산 처리에 사용된다. 각각의 분산 처리의 경우 각각 다른 컴퓨터들이 분산 처리에 사용될 수 있다. 그림 1에서는 C1, C3, C4, 그리고 C6가 하나의 작업을 위하여 분산 처리 시스템(DS1)으로 사용되고, C2, C4, 그리고 C7이 다른 하나의 분산 처리 시스템(DS2)으로 사용되고 있다. 하나의 인증 서버(CA)가 존재하여 각각의 컴퓨터의 공개키를 인증할 수 있도록 한다.

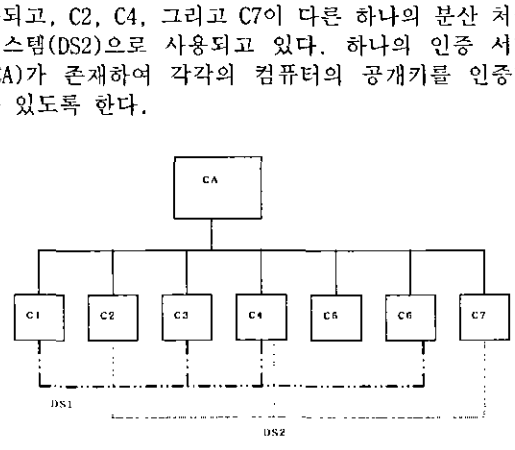

그림 1 분산 처리 시스템의 구성

3.2 키 교환

암호화는 128 비트 키 RC4 암호 알고리즘을 사용하며 무결성과 자료 근원지 인증은 HMAC-MD5가 사용되며 이들을 위한 키 공유는 일회용 Diffie-Hellman 키

교환 방식을 사용한다. 키 교환 프로토콜은 그림 2로 나타낼 수 있다. 키 교환은 매번 분산 처리 데몬이 시작될 때와 정기적(10시간)으로 수행 된다. 처음 클라이언트는 서버에게 *Hello* 메시지를 보낸다. 이 *Hello* 메시지에는 자신의 ID(IP address 사용)와 Diffie-Hellman의 공개 정보가 들어있다. 클라이언트로부터 *Hello* 메시지를 받은 서버는 클라이언트에게 *Hello* 메시지를 보낸다. 마찬가지로 이 *Hello* 메시지에는 자신의 ID와 Diffie-Hellman의 공개 정보가 들어있다. 교환된 Diffie-Hellman 공개 정보에 따라 공유된 비밀 정보가 생성된다. 공유된 비밀 정보로부터 RC4를 위한 키와 HMAC-MD5를 위한 키가 생성된다. 공유키들이 계산된 후 각각 상대방에게 *Verification* 메시지를 보낸다. *Verification* 메시지는 실제 원하는 상대방과 올바르게 키 교환이 이루어졌음을 확인하기 위하여 사용된다. *Verification* 메시지는 공유된 비밀 정보를 DSA를 이용한 전자 서명을 담고 있다. Verification 메시지의 수신자는 수신된 메시지의 전자 서명이 올바른 상대방의 전자 서명인가를 확인함으로써 올바른 상대방과 교신하고 있는가와 비밀 정보가 바르게 공유 되었는가를 확인한다. 전자 서명의 확인을 위하여 인증(CA) 서버로부터 교신하는 상대방 공개키를 확인할 필요가 있다.

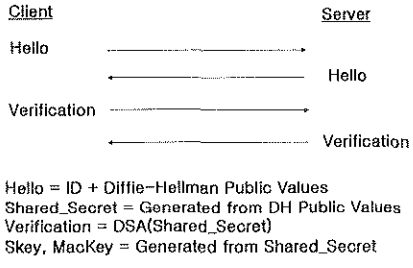

그림 2 키 교환 프로토콜

3.3 자료 송수신

키 교환이 성공적으로 종료되면 공유된 암호 키와 MAC 키를 이용하여 메시지의 암호화와 인증이 일어난다. 암호화 과정은 그림 3과 같다. 송신하고자 하는 자료로부터 공유된 MAC 키를 이용하여 HMAC-MD5를 이용하여 MAC 값이 계산된다. 송신하고자 하는 자료 뒤에 MAC을 추가한 후 공유된 암호 키를 이용하여 RC4를 이용하여 암호화 한다.

암호화 된 자료를 수신한 수신자는 공유된 암호 키를 이용하여 복호화하여 원 자료와 MAC을 복원한다. 복원된 원 자료로부터 MAC을 계산하여 수신된 MAC과 일치하는가를 검사하여 무결성과 송신자 인증을 한다.

4. 결론

비용의 증가 없이 계산 속도를 증가 시키기 위해 인터넷을 이용한 분산 처리가 제안되었다. 그러나 인터넷을 이용한 분산 처리의 경우 정보 보호의 필요성이 대두 된다. 정보 보호 기능이 있는 인터넷을 이용한 분산 처리 시스템이 개발 중에 있다. 개발 중인 시스템은 자료 교환에 기밀성을 주기 위해서는 RC4 암호 알고리즘을 사용하며 무결성과 송신자 인증을 위해서는 HMAC-MD5를 이용한다. RC4와 HMAC-MD5를 위한 키 공유를 위하여 일회용 Diffie-Hellman 키 교환 방식을 사용한다. 그리고 DSA 전자 서명을 이용하여 키 교환 시 상대방 인증을 한다. DSA 공개키 인증을 위한 인증 서버를 유지한다. 이러한 정보 보호 기능의 추가로 안전하게 인터넷을 이용한 분산 처리를 수행 할 수 있다.

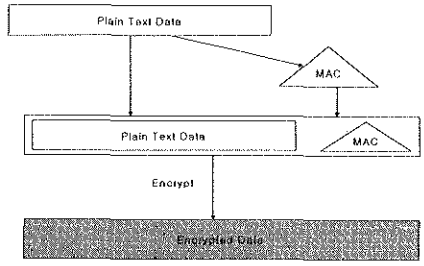

그림 3 자료 암호화 과정

5. 참고 문헌

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

[2] 송상현, 박현동, 김충길, 박중길, 김영진, 류재철, "클라이언트/서버 모델을 이용한 키 전수조사 시스템 구현," *한국정보처리학회 논문지*, 제6권 제6호, pp. 1622 - 1634, 1999.

[3] FIPS 46, "Data Encryption Standard," *Federal Information Processing Standards Publication 46*, U.S. Dept. of Commerce / National Bureau of Standards, 1977.

[4] S. Miyaguchi, "The FEAL cipher family," *Advances in Cryptology - CRYPTO' 90*, pp. 627-683, 1991.

[5] X. Lai, "On the design and security of block ciphers," *ETH Series in Information Processing*, vol. 1, 1992.

[6] B. Schneier, *Applied Cryptography*, Wiley, 1996.

[7] R. L. Rivest, "The RC5 encryption algorithm," Second International Workshop on Fast Software Encryption, pp. 86-96, 1995.

[8] *128비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서*, 한국정보보호센터, 1998

[9] R. Rivest, A. Shamir, and L. M. Adleman, "A method for digital signature and public-key cryptosystems," *Communications Of the ACM*, vol. 21, pp. 120-126, 1978.

[10] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 469-472, 1985.

[11] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, 1985.

[12] FIPS 186, "Digital signature standard," *Federal Information Processing Standards Publication 186*, U.S. Dept. of Commerce / NIST, 1994.

[13] R. L. Rivest, "The MD5 message-digest algorithm," *RFC 1321*, Apr. 1992.

[14] FIPS 180-1, "Secure hash standard," Federal Information Processing Standards Publication 180-1, U.S. Dept. of Commerce / NIST, 1995.

[15] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology - CRYPTO' 96*, pp. 1-15, 1996.