通信情報保護學會誌
第2卷, 第1號, 1992. 3

전이중 통신을 지원하기 위한 키분배 프로토콜의 분산 알고리즘 설계 및 타당성 검사

손진곤* · 허용도** · 조승한***

1. 서 론

컴퓨터 통신망의 발전은 지리적으로 분산되어 발생하는 각종 정보 및 자원을 일반 사용자들이 접근하여 사용할 수 있도록 해주었다. 그러나 컴퓨터 통신망의 사용자가 언제 어디서나 각종 정보 및 자원을 이용할 수 있게 됨에 따라 개인의 권리 침해 및 자원의 경제적 손실 등과 같은 예상하지 못한 사건들이 발생할 수 있다. 이를 방지하기 위해서 컴퓨터 통신망에 적절한 보호조치가 행해져야만 한다. 즉, 전송되는 정보의 적절한 보호조치로 정보의 불법 유출, 삭제 및 수정 등의 위험에 대처할 수 있어야 하고 허락받지 못한 사용자들로 하여금 자원을 사용할 수 없도록 하여 자원의 손실을 방지하여야만 한다[2].

정보의 보호란 컴퓨터 자체의 물리적 보호뿐 아니라 전산 요원 및 관련 시설의 보호도 포함하며 특히 통신채널을 통해 전송되는 정보가 침해자에게 노출되지 않도록 하는 것을 말한다. 따라서 컴퓨터 통신망은 자원의 보호측면과 시스템 운영상의 보호측면의 두가지 보안 정책을 항상 일관성있게 유지해야 하며, 전송되는 정보의 노출방지를 위해서 경제적이고도 효율적인 암호화 방식을 사용해야만 한다.

그러나 기존의 암호화 방식을 컴퓨터 통신망에 사용하기 위해서는 비록 암호화 알고리즘이 계산적으로 안전하다 할지라도 송수신자간의 통신키에 의한 동의가 이루어져야 통신의 완전성을 이룰 수 있다. 즉, 송신자에 의해 통신키로 암호화되어 전송된 정보는 수신자가 동일한 통신키에 의해 똑같은 정보를 획득할 수 있어야 한다. 동일한 통신키는 비밀 요원 또는 등기 우편을 이용하는 방식으로 송신자들이 안전하게 분배받을 수 있으나 이 방법은 비용이 많이 들고 시간이 지연된다는 단점을 가지고 있다. 따라서 키의 분배를 안전하면서도 자동적으로 이루기 위해 계층적 키의 사용을 통한 키분배 프로토콜(KDP : Key Distribution Protocol)이 사용된다[3]. 더우기 통신 엔티티의 인증 및 통신키의 효율적 관리를 위해 통신망 관리 요소 중 하나인 신뢰성 있는 키분배 센터(KDC : Key Distribution

* 정회원, 한국방송통신대학 전자계산학과
** 정회원, 고려대학교 전산과학과
*** 정회원, 리버티 시스템(주)

bution Center)도 요구된다[2, 11].

또한 이러한 요구 조건에 의해 개발된 KDP는 분산환경으로의 적용을 위해 전이중(full-duplex) 통신을 지원해주는 기능도 가지고 있어야 한다. 즉, client-server 관계인 반이중(half-duplex) 통신이 아닌 모든 통신 엔티티가 어느 시점에서든지 다른 통신 엔티티와 통신을 개시할 수 있다는 전이중적 명세로 변환되어질 수 있어야 한다.

이에 본 연구에서는 기존의 키분배 프로토콜을 분석하여 전이중 통신을 지원하기 위한 분산 알고리즘으로의 변환과 페트리 네트 모델을 이용한 타당성 검사를 통해 새로운 키분배 프로토콜의 개발시 프로토콜의 구현방향과 타당성 분석에 대한 지침을 제공하고자 한다.

2. 키분배 프로토콜의 고찰

2. 1. 키분배 프로토콜의 정의 및 필요성

키분배 프로토콜이란 암호화된 메세지를 교환하려는 두 통신 엔티티가 통신키에 서로 동의하는 과정을 말한다. 특히 공통키 암호화 시스템(private cryptography system)에서는 두 통신 엔티티가 한 세션에서만 사용하는 동일한 통신키에, 공개키 암호화 시스템(public cryptography system)에서는 상대방의 공개키에, 서로 동의하는 것을 의미한다. 즉, 수신자에게 송신자의 암호화 키에 대응되는 복호화 키를 분배하는 과정과 서로에 대한 상호 인증과정을 포함하는 통신 규약을 KDP라 한다.

통신하려는 두 통신 엔티티중 송신 엔티티는 한 세션의 구축시 전송될 메세지의 안전성을 위해 메세지를 통신키로 암호화하여 송신한다. 그러나 송신 엔티티가 암호화된 메세지를 수신 엔티티에게 전달하기 전에 사용된 통신키를 수신자에게 미리 알려주는 절차가 반드시 필요하다. 이를 위해서 물리적으로 안전한 통신채널을 사용할 수 있으나, 이는 지역적으로 멀리 떨어져 있는 통신망인 경우에는 유지 보수에 대한 엄청난 비용 및 그 안전성에 대한 의문이 항상 존재하므로 부적합한 방법이다.

또 다른 방식으로 비밀 요원, 등기 우편과 같은 안전성을 보장받는 통신키 분배방법을 이용할 수도 있으나, 이것도 역시 세션 구축을 위해 비교적 큰 비용 및 시간 지연을 초래하고 통신키 관리자의 고의적 또는 우발적 에러 유발의 가능성이 높아 그 안전성을 보장할 수 없다[13].

이에 대한 해결책으로 공통키 암호화 시스템에서는 통신 엔티티의 수가 n일 때 이들의 완전 그래프(complete graph) 위에서 간선의 수인 $(n-1)\times 2$ 개의 키를 서로 분배해 놓은 후 통신 엔티티 사이에 고정되어 있는 통신키를 일정 기간동안 사용하게 하는 방법이 있다. 그러나 이 방법은 비록 키관리자에 의한 키분배의 수를 현격히 줄여 시간 지연없는 통신을 이룰 수 있다 하더라도 여러 세션 동안 동일한 키를 사용함으로써 키의 노출을 용이하게 한다[10].

이 문제에 대해 Merkle[9]은 불안전한 통신 채널을 사용해 매 세션마다 하나의 통신키를 생성하며, 생성된 통신키에 두 통신 엔티티가 서로 동의하는 키분배 방법을 제시하였다. 그러나 이 방법 역시 한개의 통신키를 전송하기 위해 n개의 잠재키(potential key)가 전송되어야 하므로 높은 전송 오버헤드가 따른다.

따라서 키관리자의 키관리를 최소화하며 매 세션마다 통신키를 생성하고 분배하는데 따르는 통신 오버헤드가 적은, 자동화된(automated) 키분배 방법이 필요하게 된다[3].

공통키 암호화 시스템에서 키 분배는 매스터키(master key : 키관리자 및 각 통신 엔티티만이 가지고 있는 암호화키로 통신키를 암호화하여 전달할 때 사용함)라 불리는 키와 단순히 메세지의 암호화를 위한 통신키로 구성된 두 레벨의 계층적 키 구조를 사용한다.

한편 공개키 암호화 시스템에서는 각 통신 엔티티가 자기 고유의 비밀키(secret key) 및 공개키(public key)를 가지며, 전송하려는 메세지는 수신자의 공개키로 암호화시켜 보내면 된다. 따라서

통신망의 모든 사용자들은 상대방의 공개키를 알아내기 위해서 공개화일(public file : 통신망의 모든 사용자들의 공개키를 등록해 놓은 화일)을 검색하여야 한다. 그러나 공개키 암호화 시스템에서는 공개화일에 대한 신뢰도에 많은 문제점이 따르게 된다. 즉 침해자가 공개화일을 침범하여 저장된 정보를 파손시키는 경우 송수신자간의 완전한 통신을 보장할 수가 없게 된다.

따라서 두 통신 엔티티간의 완전한 통신을 하기 위해서는 첫째, 매 세션마다 통신키를 생성 및 분배해야 하며 둘째, 통신키의 노출이 없어야 하고 셋째, 통신 엔티티의 상호 인증을 포함하며, 넷째, 위에서 설명한 문제점들을 해결할 수 있는 키분배 프로토콜을 필요로 한다.

2. 2. 키분배 센터의 필요성

통신망 관리 요소 중 하나인 키분배 센터는 세션 구축의 요청에 대응해 통신키를 생성하고 관리하는 일종의 서버(server)로 통신망에서 사용하는 모든 통신키의 관리를 책임질 뿐 아니라 통신키의 안전한 분배를 책임진다.

만일 공통키 암호화 시스템에서 키분배 센터가 사용되지 않는다면 통신키분배 및 관리에 대한 오버헤드가 통신 엔티티에게 지나치게 편중되고, 사용된 키분배 프로토콜도 통신 엔티티의 상호 인증을 포함하지 못해, 침해자에 의한 위장을 허용할 수 밖에 없다. 공개키 암호화 시스템에서도 키분배 센터가 사용되지 않는다면 공개 화일에 대한 연산으로 첫째, 새로운 통신 엔티티가 자신의 공개키를 처음으로 등록하는 연산 둘째, 기존에 등록된 통신 엔티티의 공개키의 삭제 연산 셋째, 통신 엔티티의 공개 키 변경 요청에 따른 갱신 연산과 같은 것들이 통신망에 등록된 모든 통신 엔티티에 대해 이루어져야 하기 때문에 상당히 큰 비용과 관리의 어려움을 겪는다[4, 6, 11].

따라서 공통키 암호화 시스템에서 키분배 센터는 세션 구축에 따른 통신 오버헤드가 적으며 자동화된 통신키를 제공하기 위해 필요하고 아울러 키분배 프로토콜에 따른 상호 인증기능을 지원하기 위해서도 필요하다. 또한 공개키 암호화 시스템에서 키분배 센터는 통신하려는 통신 엔티티의 공개키를 키분배 센터 자신의 비밀키를 사용하여 제공하기 위해서 또는 공개 화일을 효율적으로 관리하기 위해서 필요하다.

이러한 키분배 센터의 운영 환경은 키분배 센터에 대한 침입이 통신망 전체의 보안 체제에 상당한 큰 영향을 미치므로 통신망 보안 정책에서 가장 상위의 안전성을 보장받아야 한다[8].

2. 3. Needham과 Schroeder의 키분배 프로토콜

Needham과 Schroeder[11]는 암호화 방식을 사용한 통신 엔티티 인증을 통해 두 통신 엔티티간의 자동화된 통신키 분배를 처음으로 제안하였다. 통신 개시자를 A, 통신 상대자를 B, A와 B에 대한 매스터키를 각각 MK_A와 MK_B라 하고 사용되는 통신키를 CK라고 약속하면, 공통키 암호화 시스템에서 Needham과 Schroeder가 제안한 키분배 프로토콜은 다음과 같다.

$$A \rightarrow KDC : A,\ B,\ EM_A \quad (1)$$

$$KDC \rightarrow A : E(MK_A : (EM_A,\ B,\ CK,\ E(MK_B : (CK,\ A)))) \quad (2)$$

$$A \rightarrow B : E(MK_B : (CK,\ A)) \quad (3)$$

$$B \rightarrow A : E(CK :\ EM_B) \quad (4)$$

$$A \rightarrow B : E(CK : f(EM_B)) \quad (5)$$

Needham과 Schroeder가 제안한 키분배 프로토콜은 3번의 메세지 교환으로 상호 통신키를 확인할 수 있으나, 통신 상대자 B가 통신 개시자 A로 부터 온 통신키 CK가 이전에 사용된 통신키인지를 확인하기 위하여 두번의 인증과정을 더 거치는 것이 그 특징이라 할 수 있다.

한편, 공개키 암호와 시스템에서의 Needham과 Schroeder의 키분배 프로토콜은 다음과 같다.

$A \rightarrow KDC : A, B$ (1)

$KDC \rightarrow A : E(SK_{KDC} : (PK_B, B))$ (2)

$A \rightarrow B : E(PK_B : (EM_A, A))$ (3)

$B \rightarrow KDC : B, A$ (4)

$KDC \rightarrow B : E(SK_{KDC} : (PK_A, A))$ (5)

$B \rightarrow A : E(PK_A : (EM_A, EM_B))$ (6)

$A \rightarrow B : E(PK_B : EM_B)$ (7)

이 키분배 프로토콜은 각 통신 엔티티가 상대방의 공개키를 갖는 것을 미리 방지하기 위해 KDC가 각 통신 엔티티의 공개키를 관리, 유지한다. 그리고 통신 엔티티들은 자신의 비밀키만을 관리한다. 또한 KDC 자신도 자기 자신을 모든 통신 엔티티에게 인증하기 위해 비밀키 SK_{KDC}와 공개키 PK_{KDC}를 가지며 이 공개키는 모든 통신 엔티티에게 공개하도록 한다.

그러나 이상과 같은 두개의 프로토콜은 이전에 사용되어 왔던 하나의 통신키가 노출되었을 경우, 침입자에 의해 침해 가능함이 Denning과 Sacco에 의해 지적되었다[2]. 즉 공통키 암호화 시스템인 경우 두 통신 엔티티의 세션 구축시 보내졌던 (3)의 메세지를 침입자가 가로채 기록해 놓았다면 그 세션이 끝난 직후 침입자가 위장된 세션 개시를 이룰 수 있다. 다시 말해서 합법적인 통신 엔티티간의 세션이 종료된 후 침입자가 (3)의 메세지를 재전송하고 (4)의 메세지를 받으면 사전 표시 EM_B에 동의 함수를 적용해 (5)의 메세지 형태로 암호화시켜 보냄으로써 침입자는 통신 개시자 A를 위장해 다시 새로운 세션을 구축할 수 있게 된다.

Denning과 Sacco는 이러한 공격에 대처하기 위해 전역 클럭(global clock)을 이용한 키분배 프로토콜을 제시하고 있으나 전역 클럭의 재동기화(resynchronization) 문제로 극히 제한된 영역에서만 사용된다는 단점이 있다.

2. 4. Otway와 Rees의 키분배 프로토콜

Otway와 Rees[4]의 프로토콜은 공통키 암호화 시스템에서의 키분배를 위한 프로토콜로 다음과 같은 4단계의 메세지 교환을 통해 통신 엔티티들이 통신키를 획득하고 상호 인증을 할 수 있게 된다.

$A \rightarrow B : A, B\ E(MK_A : (EM_A, A, B))$ (1)

$B \rightarrow KDC : A, B, E(MK_A : (EM_A, A, B)), E(MK_B : (EM_B, A, B))$ (2)

$KDC \rightarrow B : E(MK_A : (EM_A, CK)), E(MK_B : (EM_B, CK))$ (3)

$B \rightarrow A : E(MK_A : (EM_A, CK))$ (4)

이 프로토콜은 교환되는 메세지의 수가 적으면서도 Needham과 Schroeder의 프로토콜에서 발생하는 결점을 갖지 않는다. 즉 KDC가 통신키를 제공할 때 각 통신 엔티티는 자신의 사전 표시(event marker)를 이용하여 메세지의 새로움(freshness)을 검사하므로 침입자의 재전송 공격에 대해 방어적인 것이다.

그러나 위의 프로토콜도 다음과 같은 결점을 가지고 있다. 즉, (1)의 MK_A로 암호화된 메세지와 (4)의 메세지의 유사성을 통해 EM_A에 의한 디퓨징(diffusing)이 제한되어 있을 경우, 침입자는 정상적인 (3)의 메세지를 가로챈 후 (1)의 메세지중 암호화된 내용 일부를 보내면 A는 침입자도 알 수 있는 그릇된 통신키를 사용하게 된다. 이와 같은 공격에 대응하기 위해서는 (3)의 메세지 형태가 통신망의 모든 통신 엔티티에게 동의된 일정 함수 f를 적용하여 $E(MK_A : (f(EM_A, CK)$ 및 $E(MK_B : f(EM_B), CK))$로 (4)의 메세지는 $E(MK_A, (f(EM_A, CK)))$로 각각 수정되어야 한다.

Otway와 Rees가 제안한 키분배 프로토콜은 Internet 환경하에서 안전한 통신을 하기 위해 약간 개선된 형태로 제시되기도 하였다[14].

3. 전이중적 통신을 위한 KDP의 명세

3. 1. 제안된 키분배 프로토콜의 구성 및 특성

새로운 키분배 프로토콜은 통신 개시자 A가 통신 상대자 B에게 세션 구축을 요청함으로써 시작된다.

$$A \rightarrow B : A \qquad (1)$$

이때 A는 KDC에 통신키를 요구하며 B도 또한 메세지 수신과 더불어 KDC에게 통신키 요구를 A의 요구와 병행적(parallel)으로 보낸다.

$$A \rightarrow KDC : A,\ E(MK_A,\ (B,\ EM_A)) \qquad (2.1)$$
$$B \rightarrow KDC : B,\ E(MK_B,\ (B,\ EM_B)) \qquad (2.2)$$

KDC는 이러한 각각의 메세지를 수신한 후 A와 B에게 다음과 같은 메세지를 병행적으로 보낸다.

$$KDC \rightarrow A : E(MK_A,\ (EM_A,\ CK)) \qquad (3.1)$$
$$KDC \rightarrow B : E(MK_B,\ (EM_B,\ CK)) \qquad (3.2)$$

위의 프로토콜을 도식화하면 그림 1과 같아진다.

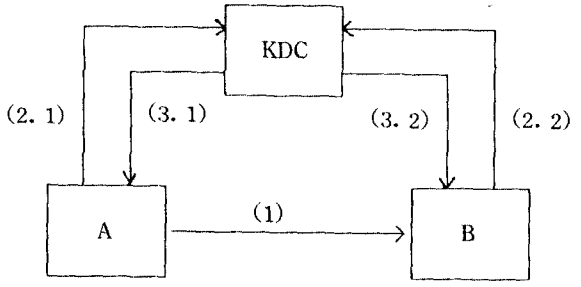


그림 1. 제안된 키분배 프로토콜

위에서 설명한 키분배 프로토콜은 기존의 키분배 프로토콜과 비교하여 어떠한 침해자의 위장도 허용되지 않음이 위장 탐지 알고리즘[15]으로부터 분석되어진다. 제시된 키분배 프로토콜은 기존의 다른 키분배 프로토콜과 비교할 때 다음과 같은 특성을 갖는다[16].

1) A와 B에 의해 (2.1)과 (2.2)의 메세지가 각각 KDC에게 독립적으로 보내질 때 (2.1)의 KDC 수신은 (3.1)의 전송을, (2.2)의 KDC 수신은 곧바로 (3.2)의 전송을 유발하므로 그 병행성을 통해 키분배 지연을 단축시킬 수 있다.

2) KDC의 설계측면에서 (2.1)과 (3.1)에 대한 (2.2)와 (3.2)의 대칭성은 KDC의 처리절차를 단순화시킨다.

3) 그림 1과 같이 제시된 키분배 프로토콜은 암호화 시스템의 변경에 적응성있는 프로토콜로 채택한 시스템이 어느 암호화 방식을 채택하더라도 프로토콜 자체의 변경이 아닌 암호화 키 매개변수의 변경만으로 키분배를 다음과 같이 안정성있게 할 수 있다. 즉 공통키 암호화 시스템인 경우에는 (a)처럼, 공개키 암호화 시스템인 경우에는 (b)처럼, 공개키 인증을 사용한 공통키 암호화 시스템에서는 (c)와 같이 쉽게 변형이 된다.

(a) 공통키 암호화 시스템인 경우

$$A \rightarrow B : A \qquad (1)$$
$$A \rightarrow KDC : A,\ E(MK_A,\ (B,\ EM_A)) \qquad (2.1)$$
$$B \rightarrow KDC : B,\ E(MK_B,\ (B,\ EM_B)) \qquad (2.2)$$

KDC→ A : $E(MK_A, (EM_A, CK))$ (3.1)
KDC→ B : $E(MK_B, (EM_B, CK))$ (3.2)

(b) 공개키 암호화 시스템인 경우

A → B : A (1)
A → KDC : A, $E(SK_A, (B, EM_A))$ (2.1)
B → KDC : B, $E(SK_B, (B, EM_B))$ (2.2)
KDC→ A : $E(SK_{KDC}, (EM_A, PK_B))$ (3.1)
KDC→ B : $E(SK_{KDC}, (EM_B, PK_A))$ (3.2)

(c) 공개키 인증을 사용한 공통키 암호화 시스템

A → B : A (1)
A → KDC : A, $E(SK_A, (B, EM_A))$ (2.1)
B → KDC : B, $E(SK_B, (B, EM_B))$ (2.2)
KDC→ A : $E(SK_{KDC}, (EM_A, CK))$ (3.1)
KDC→ B : $E(SK_{KDC}, (EM_B, CK))$ (3.2)

3. 2. 제안된 키분배 프로토콜의 분산 알고리즘

본 절은 3. 1절에서 제시한 키분배 프로토콜에 대한 구체적인 내용의 설명으로 통신 엔티티들 사이에서 교환되는 제어메세지의 내용과 전이중 통신을 지원하기 위한 키분배 프로토콜의 분산알고리즘을 설명한다.

3. 2. 1. 제어 메세지의 교환

각 통신 엔티티가 세션 구축을 제어하기 위해 갖는 지역 변수는 다음과 같다.

(1) request : 통신 엔티티들의 식별자(id)들의 집합으로 처음에는 공집합으로 할당됨
(2) acceptance : 통신 엔티티들의 식별자와 사건표시기(EM)의 집합으로 처음에는 공집합으로 할당됨
(3) handshake1 handshake2 : 논리형 값을 갖는 배열로서 크기는 통신 엔티티의 수이며 처음에는 거짓으로 할당됨

지역변수 request는 세션 구축을 요청하거나(issue) 요청받은 경우에 상대 통신 엔티티를 포함하는 집합을 나타낸다. 지역변수 acceptance는 KDC에 세션키를 요구한 후 KDC로부터의 메세지가 자신의 요청에 의해서 보내어 진 것인지, 즉 재전송에 의해 보내진 것이 아님을 보장하는 수락 조건 검사를 위한 집합이다. 배열 handshake1, handshake2는 통신하려는 상대 통신 엔티티가 KDC로부터 세션키를 받았음을 자신이 인식하기위한 지역 변수이다.

한편 KDC는 통신하려는 두 통신 엔티티가 서로 동일한 세션키를 지니도록 해주기 위해서 통신 엔티티들이 갖는 것과는 다른 지역변수 handshake를

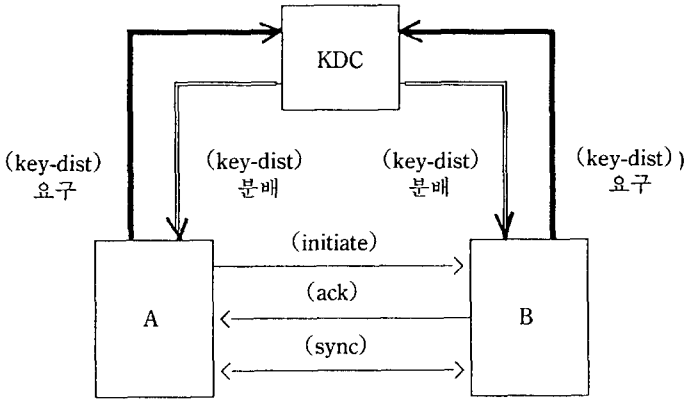


그림 2. 제어 메세지의 교환

갖는다.

handshake : 통신개시자, 통신상대자 및 통신키의 집합

이상에서 설명한 지역변수를 이용하여 KDP의 구현을 위해 사용되는 메세지의 형태들은 다음과 같다.

(1) (initiate, id) : 세션 구축을 요구

(2) (ack, id) : 세션 구축 요구에 따른 응답 메세지

(3) (key-dist, message) : KDC에게 세션키를 요구 또는 KDC로 부터 세션키를 수신

(4) (sync, id) : 세션 확립

이러한 제어 메세지 교환은 그림 2에 잘 나타나 있다. 여기에서 두 종류의 굵은 실선들은 그들이 각각 병행적으로 수행되는 것을 표시한다.

3. 2. 3. 전이중 통신을 위한 분산 알고리즘

3. 1절에서 제시한 키분배 프로토콜을 3. 2. 1절에서 제시한 지역변수와 메세지를 이용하여 분산 알고리즘으로 표현하면 아래 알고리즘과 같다.

이 알고리즘에서 when절은 사건의 발생에 따른 대응 연산(event-driven operation)을 위한 것이다. 이것은 프로그래밍 언어 Ada에서 exception handling 부분에 해당한다고 볼 수 있다. 따라서 그 운영 방식은 순차적으로 사건을 처리하는 모니터(monitor) 구조를 따를 수도 있고, 또는 사건마다 이 코드를 공유하며 처리되어도 무방하다.

알고리즘에서 (1)은 현재 세션키의 분배를 위해 처리중임을 나타내기에 특정 코드가 제시되지 않았으며 (3)에 의한 연결의 완료시 호출시켜 통신할 수 있게 하면 된다. 그리고 (2)는 두 통신 엔티티가 동시에 세션 구축을 서로에게 요청한 경우의 처리를 위한 것으로서 어느 한 통신 개신자에 대한 키분배 프로토콜의 분배일 경우에도 적용이 된다. 그러나 여기서는 통신 엔티티 식별자(identifier)가 큰 값을 갖는 통신 개시자에 의한 요청만을 처리하도록 한다. 따라서 작은 식별자 값을 갖는 세션 요청의 메세지는 무시된다.

ALGORITHM

```
(* Algorithm for communication entity A *)

When wish to communicate with B do
  begin
    if B ∈ request then "connection processing" /* (1) */
        else begin
                request:=request ∪ {A};
                send (initiate, A) to B;
             end
  end;

when receive (initiate, B) do
  begin
    if (B<A) & (B ∈ request) then "disregard" /* (2) */
        else begin
```

```
                request:=request ∪ {B};
                EMA:=Event-Marker;
                send (ack, A) to B:
                send (key-dist, A, E(MKA, (B, EMA))) to KDC;
                acceptance:=acceptance ∪ {(B, EMA)}
              end
  end;

when receive (ack, B) do
  begin
    EMA:=Event-marker;
    send (key-dist, A, E(MKA, (B, EMA))) to KDC;
    acceptance:=acceptance {(B, EMA)}
  end;

when receive (key-dist, m) do
  begin
    (B, EM, CK):=D(MKA, m);
    Communication-Key-Table(B):=E(MKA, CK);
    if (B, EM) ∈ acceptance
      then begin
             send (sync, A) to B;
             handshakel(B):=true;
             if handshake2(B) then "connection complete"/*(3) */
           end
  end;
when receive (sync, B) do
  begin
    if (B, −) ∈ acceptance
      then begin
             send (sync, A) to B;
             handshakel(B):=true;
             if handshake2(B) then "connection complete" /*(3)*/
           end
  end;

when terminate to communication with Q do
  begin
    request:=request−{B};
    acceptance:=acceptance−{(B, −)}
    handshakel(B):=false;
    handshake2(B):=falese;
  end;

(* algorithm for key distribution center *)
```

```
when receive (key-dist, A, m) do
  begin
    (B, EM):=D(MK_A, m);
    if ∃ h ∈ handshake such that(h1=A & h2=B)
                                  or (h1=B & h2=A)
      then begin
             send (key-dist, E(MK_A, (B, EM, h3))) to A;
             handshake:=handshake-{h}
           end
      else begin
             CK:=Key-Generator;
             send (key-dist, E(MK_P, (B, EM, CK))) to A;
             handshake:=handshake {(A, B, CK)};
           end
  end;
```

3. 3. 페트리 네트를 이용한 타당성 검사

본 절에서는 3. 2. 2절에서 제시한 분산 알고리즘에 대해 억제 아크(inhibitor arc)를 이용한 페트리 네트 모델로 모델링한 후 그 타당성을 검증하였다[7, 15, 16].

분산 알고리즘 1의 타당성 분석을 위해 특정 통신 엔티티의 페트리 네트모델이 그림 3의 (a)에 잘 나타나 있다. 특히 트랜지션(transition) SA^1과 SA^2에서는 억제 아크를 사용한 것이 보이는데 그 이유는 다음과 같다. 첫째, 트랜지션 SA^1에 의한 응답(ack) 메세지 송신은 통신에 대한 요청 및 통신 개시에 대한 메세지 전송이 없을때, 다른 통신 엔티티에 의해 보내어진 통신 개시 메세지에 대해 응답 메세지를 보내는 상태 전이를 나타내기 위한 것이다. 둘째, 트랙잭션 SA^2는 통신에 대한 요청은 있었으나 아직 통신 개시 메세지를 보내지 않은 상태일 때, 다른 통신 엔티티에 의해 보내어진 통신 개시 메세지에 대한 응답 메세지를 먼저 처리하도록 하는 상태 전이를 나타내기 위한 것이다.

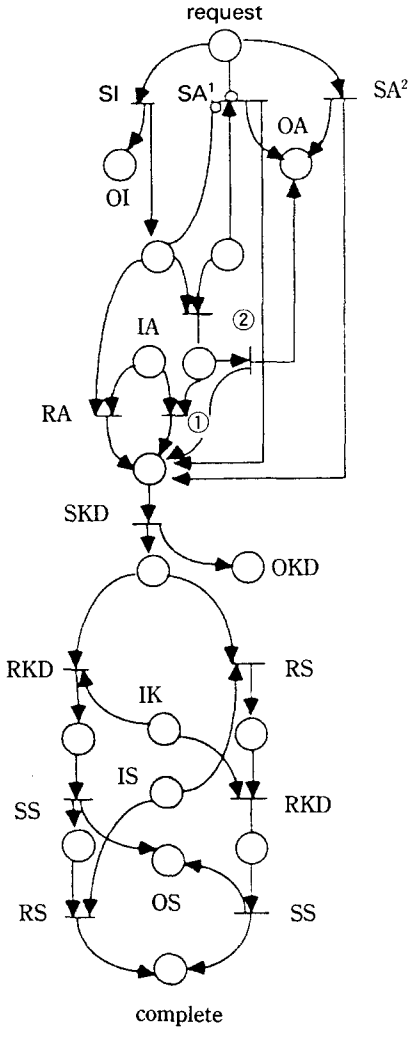
또한, 각 프레이스(place)들의 역할과 트랜지션(transition)의 점화(fire)에 대한 설명은 그림 3의 (b)에 나타나 있으며, 페트리 네트 모델에 대한 블럭 형태로의 단순화는 그림 3의 (c)에 잘 나타나 있다.

두개의 통신 엔티티 A, B에 대해 어느 하나 또는 둘 모두가 세션 구축을 요청할 경우, 세션키 분배를 위한 알고리즘 1의 전체 시스템 상태 전이를 나타내는 페트리 네트 모델은 그림 4와 같이 블럭된 형태로 나타낼 수 있다.

블럭된 페트리 네트 모델에서 두개의 통신 엔티티 A, B에 대해 A의 식별자 값이 B의 식별자 값보다 작을 경우, A의 페트리 네트 블럭에서는 ①의 트랜지션이, B의 페트리 네트 블럭에서는 ②의 트랜지션이, 점화가 가능하지 않은 트랜지션이 된다 (그림 3의 (a)참조).

따라서 그림 4의 페트리 네트 모델을 통해 어느 한 통신 엔티티가 세션구축을 요청하거나 두개의 통신 엔티티가 동시에 세션 구축을 요구하는 경우에도 키분배에 의한 세션이 확실히 구축된다는 성질이 보장된다. 그 이유는 세션 요청 플레이스가 마크된 상태에서 도달 가능성 트리 생성자(reachability tree generator) 알고리즘에 의해 생성되는 도달가능성 트리로부터 확인할 수 있다[17]. 즉, A

또는 B의 세션 구축 요청 플레이스가 하나만 마크되거나 A와 B의 세션 구축 요청 플레이스가 둘다 마크된 상태일 때도 어떠한 데드록이 발생하지 않고 A와 B의 두개의 complete 플레이스가 각각 하나의 토큰을 갖게되기 때문이다.

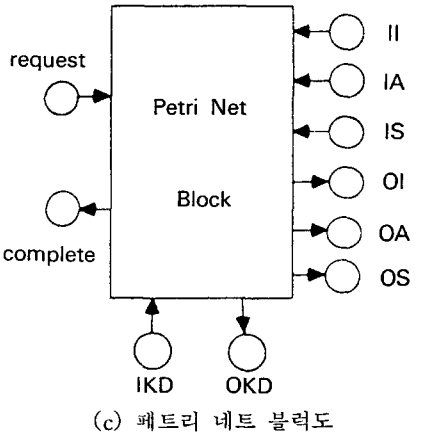


(a) 페트리 네트 구성도

Transition
SI : send initiate message
SA : send ack message
SKD : send key-dist message
SS : send sync message
RA : receive ack message
RKD : receive key-dist message
RS : receive sync message
① : fire only when its id is greater than initiator
② : fire only when its id is less than initiator

Place
request : communication issue
complete : session completion
II : input place of initiate
IA : input place of ack
IS : input place of sync
IKD : input place of key-dist
OI : output place of initiate
OA : output place of ack
OS : output place of sync
OKD : output place of key-dist

(b) transition 및 place의 역할



(c) 페트리 네트 블럭도

그림 3. 알고리즘 (제3. 2. 3절)에 대한 페트리 네트

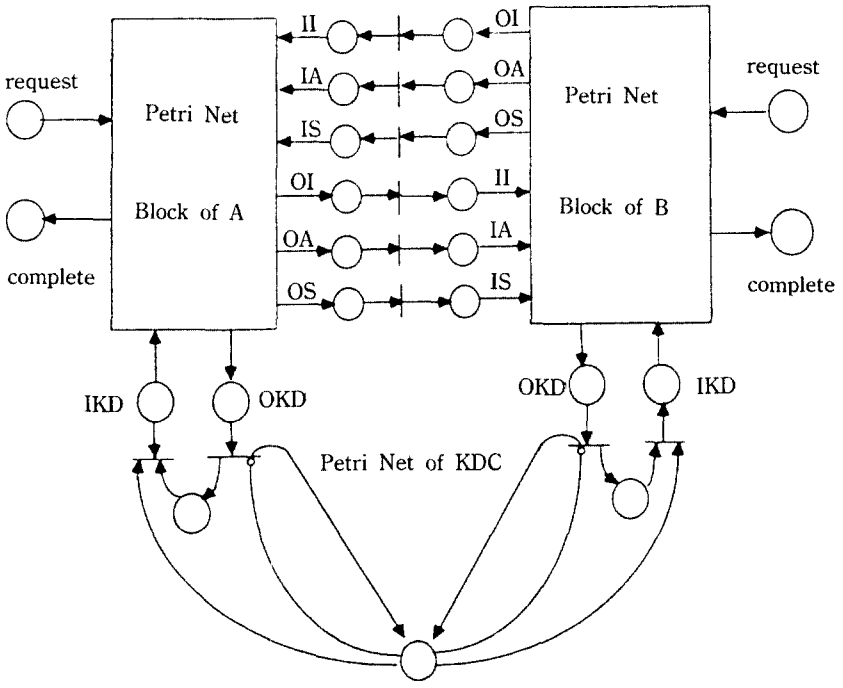


그림 4. 세션 구축을 위한 페트리 네트

4. 결 론

컴퓨터 통신망에서 키분배 프로토콜의 설계는 안전성에 관한 오류가능성이 높은 작업이어서 그 명세의 단순화를 요구한다. 또한 모든 통신 엔티티가 어느 시점에서든지 다른 통신 엔티티와 통신을 개시할 수 있다는 전이중 통신을 위한 명세로의 변환도 필요하다고 문헌[1,5]에서 설명하고 있으나 실제로 변환된 것은 없다.

이러한 변환은 Kerberos[6]의 반이중 통신 개시를 전이중 통신 개시로 확장하기 위해, Sidhu[5]의 전이중적 통신 개시의 타당성 분석을 체계화시키기 위해서 사용될 수 있다.

따라서 본 연구에서는 기존의 키분배 프로토콜을 분석하여 전이중 통신을 지원하기 위한 키분배 프로토콜을 제시하였다. 또한 제시된 키분배 프로토콜은 암호화 시스템의 변경에 적응성있는 프로토콜로서, 채택한 시스템이 어느 암호화 방식을 채택하더라도 프로토콜 자체의 변경이 아닌 암호화 키 매개변수의 변경만으로 키분배를 안전성있게 할 수 있다. 즉 암호화 방식이 공통키를 사용하든지 공개키를 사용하든지에 관계없이 두 경우를 모두 수용할 수 있는 프로토콜인 것이다.

아울러 본 논문에서 제시한 키분배 프로토콜에 대한, 전이중 통신을 위한 명세로서 분산 알고리즘을 개발하였으며, 페트리 네트 모델을 이용하여 이 알고리즘이 데드록이 발생하지 않는다(deadlock free)는 사실을 분석하였다. 이러한 결과로부터, 앞으로 키분배 프로토콜을 개발할 때 프로토콜의 구현방향과 타당성 분석에 대한 지침을 제공받을 수 있을 것이다.

참 고 문 헌

1. D. Dolev and A. C. Yao, "On the Security of Public Key Protocols", *IEEE Trans. Inforamation Theory*, Vol. 29, No. 2, pp. 198-208, Mar. 1983.

2. D. E. Denning and G. M. Sacco, "Timestamps in Key Distribution Protocols", *Comm. ACM*, Vol. 24, No. 8, pp. 533-536, Aug. 1981.

3. D. M. Balenson, "Automated Distribution of Cryptography Keys Using The Financial Institution Key Management Standard," *IEEE COMM. MAGAZINE*, pp. 387-392, Sep. 1985.

4. D. Otway and O. Rees, "Efficient And Timely Mutual Authentication", *Operation System Review*, Vol. 21, No. 1, pp. 8-10, Jan. 1987.

5. D. P. Sidhu, "Authentication Protocols for Computer Networks : I", *Computer Networks and ISDN Systems*, Vol. 11, pp. 297-310, 1986.

6. J. G. Steiner, C. Neuman, J. I. Schiller, "Kerberos : An Authentication Service for Open Network Systems", *Proc. the USENIX 1988 Winter Conf.*, pp. 191-202, 1988.

7. J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

8. M. Raynal, *Distributed Algorithms and Protocols*, John Wiley & Sons, 1988.

9. R. C. Merkle, "Secure Communications over Insecure Channels", *Comm. ACM*. Vol. 21, No. 4, pp. 146-151, April 1987.

10. R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures And Public Key Cryptosystems", *Comm. ACM*. Vol. 21, No. 2, pp. 120-126, Feb. 1978.

11. R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication Large Networks of Computers," *Comm. ACM*. Vol. 21, No. 12, pp. 993-999, Dec. 1978.

12. V. L. Voydock and S. T. Kent, "Security Mechanisms in High level Network Protocols", *ACM Computing Surveys*, Vol. 15, No. 2, June 1983.

13. W. Diffie and M. E. Hellman, "New Directions in Cryptography", *IEEE Trans. Information Theory*, Vol. IT-22, No. 6, pp. 135-144, Nov. 1976.

14. W. P. Lu and M. K. Sundareshan, "Secure Communication in Internet Environments : A Hierachical Key Management Scheme for End-To-End Encryption", *IEEE Trans. Comm.*, Vol. 37, No. 10, pp. 1014-1023, Oct. 1989.

15. 조승한, 황종선, "보완된 안전성 분석 방법에 의한 키분배 프로토콜의 개발", 고려대학교 석사논문, 1991.

16. 손진곤, 조승한, "효과적인 키 분배 프로토콜의 개발 및 보안성 검증", 한국방송통신대학 논문집, Vol. 13, pp. 367-38, 1991.

17. 유헌창, "마킹 그래프에 의한 Petri net 모델의 교착상태 탐색에 관한 연구", 고려대학교 석사논문, 1990.

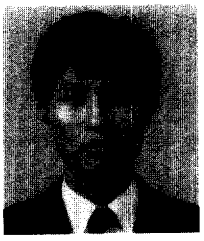
□ 著者紹介

孫 進 坤(正會員)

1984년 고려대학교 수학과 졸업(이학사)
1988년 고려대학교 대학원 수학과 전산학전공(이학석사)
1991년 고려대학교 대학원 수학과 전산학전공(이학박사)
1991년~현재 한국방송통신대학 전자계산학과 조교수
관심분야 : 컴퓨터 통신망, 분산처리 시스템, 모델링과 시뮬레이션, 컴퓨터 시큐리티 등.

許 庸 道(正會員)

1986년 고려대학교 수학과 졸업(이학사)
1988년 고려대학교 대학원 수학과 전산학전공(이학석사)
1992년 고려대학교 대학원 수학과 전산학전공 박사과정 수료
1992년~현재 : 건양대학 전자계산학과 강사
관심분야 : 분산처리 시스템, 컴퓨터 시큐리티 등.



趙 承 漢(正會員)

1989년 고려대학교 수학과 졸업(이학사)
1992년 고려대학교 대학원 전산과학과 전산학전공(이학석사)
1992년~현재 : 리버티 시스템(주) 연구원
관심분야 : 분산처리 시스템, 컴퓨터 시큐리티 등.