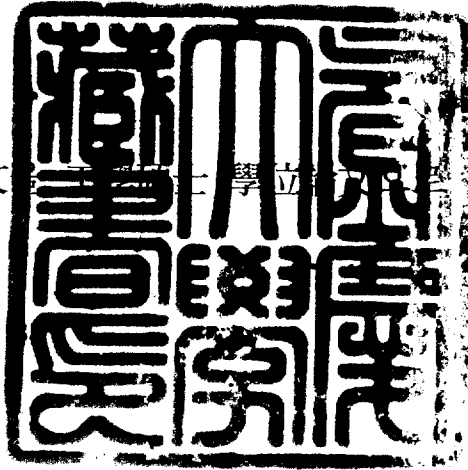


공학석사 학위논문

이기종 네트워크를 위한 적응적
네트워크 보안 관리

지도교수 정 목 동

이 論文 提出함



2005년 2월

부경대학교 대학원

컴퓨터공학과

채 종 우

채종우의 공학석사 학위논문을 인준함

2004년 12월 일

주 심 공학박사

조 우 현



위 원 공학박사

이 경 현



위 원 공학박사

정 목 동



목 차

요약	v
Abstract	vii
제 1 장 서 론	1
제 2 장 관련 연구	3
2.1 MAUT(Multi-Attribute Utility Theory)	3
2.1.1 유틸리티 함수 선정 과정	3
2.2 간결한 휴리스틱스	6
2.3 SNMP(Simple Network Management Protocol)	8
2.4 네트워크 보안 관리와 네트워크 관리	8
2.5 기존 보안 모델의 정적인 정책 결정	11
제 3 장 적응적 네트워크 보안 관리 모델	13
3.1 적응적 보안 등급 알고리즘	13
3.2 보안 정책과 접근 정책	18
3.3 ASMIB(Adaptive Security MIB)	19
3.4 전체 시스템 구조	20
3.5 Proxy에 의한 NSMS의 계층적 분산	24
3.6 적응적 보안 등급의 동적인 적용	26
3.6.1 인증 기법의 동적인 적용	26
3.6.2 대칭 암호화 기법의 동적인 적용	29
3.7 적응적 네트워크 보안 관리 모델의 보안 관리 절차	30
제 4 장 구현 및 평가	32
4.1 적응적 보안 등급 알고리즘과 보안 모듈	32
4.1.1 평가	33
4.2 SNMP와 이동 에이전트 기반 적응적 네트워크 보안관리 시스템	37

4.2.1 평가	37
4.3 기존 보안관리 시스템과의 비교	39
제 5 장 결론	40
참고문헌	41

그림 목 차

[그림 1] 두 특징 변수 결과 공간	4
[그림 2] SNMP 기반 모델과 이동 에이전트 기반 모델의 수행방식 비교	10
[그림 3] 혼합 모드의 수행방식	11
[그림 4] 적응적 보안등급을 적용한 SSL/TLS Handshake	12
[그림 5] Context Engine과 Security Context Bucket	21
[그림 6] SNMP Trap을 사용한 혼합모드	22
[그림 7] 전체 시스템 구조의 예	23
[그림 8] Proxy	25
[그림 9] Proxy에 의한 NSMS의 계층적 분산	25
[그림 10] One Time Password System의 인증 과정	28
[그림 11] 적응적 네트워크 보안 관리 모델의 보안 관리 절차	30
[그림 12] 구현 시스템	32

표 목 차

[표 1] 인증기법과 프로토콜	15
[표 2] 대칭 암호 알고리즘과 운영모드	15
[표 3] 공개키 암호 알고리즘과 해쉬 알고리즘	15
[표 4] SecurityLevel(securityProblem) 함수	16
[표 5] MAUT(X) 함수	16
[표 6] TakeTheBest($u(x_1, x_2, \dots, x_n)$) 함수	17
[표 7] GetUtilFunction(x_i) 함수	17
[표 8] 보안 정책의 예	18
[표 9] 환경 변수의 환산표	19
[표 10] 접근 정책의 예	19
[표 11] 비밀키 생성과정	29
[표 12] 보안 정책	33
[표 13] 환경 변수의 환산표	34
[표 14] 접근 정책	34
[표 15] 인증기법과 해쉬 알고리즘	35
[표 16] 대칭 암호 알고리즘과 운영모드	35
[표 17] 키 생성과 대칭암호의 응답시간 비교	36
[표 18] 인증기법들의 응답 시간 비교	37

이기종 네트워크를 위한

적응적 네트워크 보안 관리

채 종 우

부경대학교 대학원 컴퓨터공학과

요약

기존의 보안 모델들은 대부분 보안 관리를 위한 정책들을 결정하기 위하여 정적인 의사결정 방법을 따른다. 현재의 컴퓨팅 환경은 전송 속도, 전송 매체, 연결성, 대역폭 등과 같은 속성들이 상이한 이기종 네트워크 환경들이 혼재하며 이러한 속성들은 동적으로 변화한다. 더구나 다양한 연산 능력과 기능을 지닌 장비들이 널리 사용되고 있다. 이러한 네트워크 환경을 안전하게 보호하기 위해서는 다양한 네트워크 환경의 변화에 따라서 여러 가지 보안 등급을 동적으로 적용할 수 있어야 한다.

이에 본 논문에서는 네트워크 환경의 다양한 변화에 따라 보안 정책과 서비스 정책을 동적으로 적용할 수 있는 적응적 네트워크 보안 모델을 제안한다. MAUT와 간결한 휴리스틱스를 사용하여 지능적 의사결정을 수행하고

다양한 네트워크 환경을 관리하기 위하여 SNMP를 사용한다. 또한 자율적인 동작과 과도한 네트워크 트래픽을 줄이기 위하여 이동 에이전트를 사용한다.

따라서 제안된 모델은 순수 SNMP 기반의 모델에 비하여 사용자와 시스템 사이의 상호 작용을 줄일 수 있을 것이다. 더구나, 기존의 접근 방법에 비하여 다양한 변수들을 고려할 수 있으므로 이 모델은 이기종의 네트워크 환경에 대하여 보다 유연한 보안 관리를 제공할 수 있을 것이다.

Adaptive Network Security Management for Heterogeneous Networks

Jong Woo Chae

Dept. of Computer Eng., Graduate School,

Pukyong National University

Abstract

Traditional security models usually work according to a static decision-making approach to determine policies for security management. In the current heterogeneous network environment, however there are many different properties to consider in the security management such as transmission speed, communication media, connectivity, bandwidth, and so on, whose properties are dynamically changing. Moreover many types of computing devices are widely used and they have diverse capabilities in this environment. To secure this diverse environment, we

should adapt several security levels dynamically according to the changes of diverse network environment.

In this thesis, we develop an adaptive network security management model, which can dynamically adapt policies according to the changes of diverse and dynamic network environments. We use MAUT and Simple Heuristics to achieve intelligent decision making, and use SNMP to deal with diverse network environment. Also we utilize Mobile Agent for autonomous behaviors and reduction of excessive network traffic.

Thus, the proposed model might reduce the interactions between user and system in comparison with the pure SNMP based model. Moreover, the proposed model deals with multiple variables instead of single variable in the existing approach. Therefore, this model is expected to provide more flexible security management in the heterogeneous network environments.

1. 서론

오늘날의 컴퓨팅 환경은 전송매체, 대역폭, 단말의 유형, 네트워크의 구성이 다른 이기종의 네트워크들이 혼재하며 데스크톱 PC, PDA, 휴대전화와 같은 다양한 기능과 성능을 가진 컴퓨팅 단말들이 사용되고 있다. 그런데 이러한 다양한 이기종 네트워크 환경들은 동적으로 변화하므로 이와 같은 이기종 네트워크 환경을 안전하게 보호하기 위해서는 환경에 보다 적응적인 보안 관리가 필요하다.

기존의 보안 모델은 정책 결정에 정적인 의사결정 방법을 사용함으로써 환경의 변화에 적절히 대응하기 힘들며, 효율성이 부족하다. 예를 들면, 기존의 SSL/TLS는 Client Hello 단계에 클라이언트에서 전송된 Cipher Suite List에서 가장 보안 강도가 높은 Cipher Suite를 선택한다[16, 17]. 하지만 이러한 정적인 의사결정 방법은 자원이 열악한 사용자 단말에는 과부하와 지나치게 긴 대기시간을 초래할 수 있으며 암호 알고리즘은 대부분 컴퓨팅 자원을 많이 소모하므로 서버 측에도 과부하를 초래할 수 있다. SSL/TLS Handshake 과정에 적응적 보안 모델을 적용함으로써 이러한 문제점들을 해결할 수 있다[15]. Client Hello 단계에서 서버는 다양한 변수를 고려하여 동적으로 정책을 결정하는 적응적 보안 모델에 따라 Cipher Suite를 선택한다. 환경이 변화하여도 기존의 SSL/TLS는 가능한 보안 강도가 높은 Cipher Suite를 선택하는데 비해 적응적 보안 모델에 의하면 환경에 적절한 Cipher Suite를 선택할 수 있다.

따라서 본 논문에서는 다양한 네트워크 환경의 변화에 따라서 동적으로

정책을 적용할 수 있는 적응적인 네트워크 보안 관리 모델을 제안한다. 이 모델은 다중 변수 기반 유틸리티 이론인 MAUT (Multi-Attribute Utility Theory)[1, 2]와 훈련 집합의 크기가 작은 경우에 효율적인 의사결정을 수행하는 간결한 휴리스틱스(Simple Heuristics)[3]를 사용하여 지능적인 의사결정을 수행하며 다양한 네트워크 노드에 대하여 관리를 수행할 수 있는 프로토콜인 SNMP(Simple Network Management Protocol)[4]를 사용하여 다양한 네트워크 환경을 지원한다. 또한, 자율적인 동작과 보안 관리를 위한 네트워크 트래픽을 줄이기 위하여 이동 에이전트를 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대하여 기술하고 3장에서는 이기종 네트워크 환경에 대한 적응적 네트워크 보안 모델을 설명한다. 4장에서는 구현 및 평가에 대하여 설명하고 5장에서는 결론과 함께 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 MAUT(Multi-Attribute Utility Theory)

MAUT는 다중 변수에 대한 의사 결정 문제(decision problem)에서 유틸리티(utility)를 통한 정량적인 의사 결정 방법이다. 유틸리티 분석(utility analysis)은 의사 결정자(decision maker)가 원하는 제비뽑기(lottery)의 결과(consequence)를 분석 해주는 분야로서 의사 결정자는 이들 결과에 대한 개인의 선호도(preference)를 유틸리티 수(utility number)로 표현하고 있다. 결국 유틸리티는 0과 1사이의 상대적인 값으로서 $u(x^o)$ 와 $u(x^*)$ 를 각각 가장 선호하지 않는 결과 유틸리티와 가장 선호하는 결과 유틸리티라고 두면 결과 유틸리티의 최소값은 $u(x^o) = 0$, 최대값은 $u(x^*) = 1$ 로 나타낼 수 있다. 결과에 대한 유틸리티 수의 대입은 기대 유틸리티 (expected utility)를 최대화 시켜주는 쪽으로 이루어져야한다. 즉, 기대 유틸리티의 최대화는 의사 결정자의 최적 행동의 기준이 된다. 유틸리티 함수를 선정하는 데 일률적인 방법이 있는 것은 아니지만 공통적으로 사용될 수 있는 과정은 대체로 다음과 같다 [1, 2].

2.1.1 유틸리티 함수 선정 과정

선정 준비: 결과 Q 를 실수 x 에 사상시키는 평가 함수를 \mathbf{X} 라고 하면, $x = \mathbf{X}(Q)$ 이다. x 값의 크기와 바람직한 정도와의 관계를 정할 수 있다. 즉,

의사 결정자가 결과 x_1 과 결과 x_2 중에서 어떤 것을 선호하는지 확인해 볼 수 있다. 그림 1은 두 개의 특징 변수(two-attribute) Y 와 Z 에 대한 결과 공간을 보여주고 있는데, 결과 T 와 결과 S 중에서 의사결정자가 선호하는 것을 질의를 통해서 확인해볼 수 있다. 결과 Q 는 $y = y_1$ 이고 $z = z_1$ 인 결과이고, 결과 R 은 $y = y_2$ 이고 $z = z_2$ 인 결과이다.

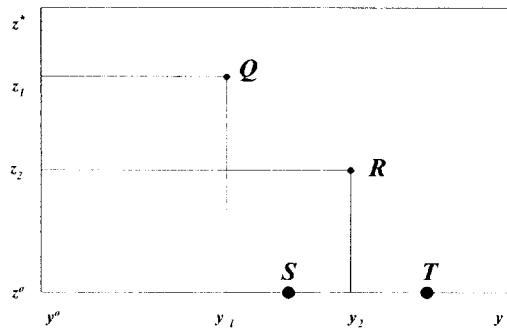


그림 1. 두 특징 변수 결과 공간

독립성 확인: Y 와 Z 가 덧셈 독립(additive independence), 유틸리티 독립(utility independence)인지 확인 해본다. Y 와 Z 가 덧셈 독립이라는 의미는 임의의 제비뽑기를 두 번 했을 때 상호 선호도 비교(paired preference comparison)는 이들의 한계 확률 분포(marginal probability distributions)에만 의존하고, 결합 확률 분포(joint probability distributions)와는 무관하다는 것을 의미한다. 즉 Y 와 Z 가 상호 독립이라는 것이다. Y 와 Z 가 유틸리티 독립이라는 의미는 z 가 주어졌을 때 Y 에 대한 제비뽑기의 조건 선호(conditional preferences)가 Z 값의 변화와는 무관하다는 것이다. 이는 Y 의 값의 변화에 따른 Y 에 대한 선호도의 변화가 Z 의 변화와는 관계없다는 의미이

다.

정성적인 성질 확인: 유틸리티 함수가 단조(monotonic) 함수인지 확인한다. x_k 가 x_j 보다 크면 x_k 는 항상 x_j 보다 좋은 것인지 확인하고, 다음으로 유틸리티 함수 u 가 모험 회피(risk averse), 모험 중립(risk neutral), 모험 노출(risk prone) 중에서 어떤 것인지 결정한다. 우선 임의의 x 와 h 에 대해서 의사 결정자의 선호도를 확인하는데, 제비뽑기 $\langle x+h, x-h \rangle$ 와 기대결과(expected consequence) x 중에서 어떤 것을 좋아하는지 확인한다. 제비뽑기 $\langle x+h, x-h \rangle$ 는 같은 확률로서 $x+h$ 와 $x-h$ 를 선택할 수 있다는 것을 의미한다. 이런 시험을 여러 다른 x 와 h 값에 대해서 반복한 결과[18], 만약 의사 결정자가 주로 제비뽑기를 선호하면 모험 노출이라고 추정할 수 있고, 기대 결과 x 를 선호하면 모험 회피라고 간주 할 수 있다. 제비뽑기와 기대 결과 사이에 특별한 선호가 없으면 모험 중립이라고 본다.

유틸리티 함수 결정: 간단한 예로 의사 결정자의 유틸리티 함수가 x 에서 단조 증가이고, 감소적인 모험 회피라고 가정하면, 이러한 특징을 만족시키는 유틸리티 함수는 다음과 같다.

$$u(x) = h + k(-e^{-ax} - be^{cx}), \text{ 여기서 } a, b, c, k \text{는 양의 상수이다.}$$

일반적으로 유틸리티 함수 $u(x_1, x_2, \dots, x_n)$ 가 덧셈 독립, 유틸리티 독립이면 u 는 다음과 같다.

$u(x_1, x_2, \dots, x_n) = k_1u_1(x_1) + k_2u_2(x_2) + \dots + k_nu_n(x_n)$, 여기서 모든 i 에 대해서 $u_i(x_i^o) = 0$, $u_i(x_i^*) = 1$.

$u_i(x_i^o)$ 와 $u_i(x_i^*)$ 는 각각 가장 선호하지 않는 결과 유틸리티와 가장 선호하는 결과 유틸리티이다.

2.2 간결한 휴리스틱스

간결한 휴리스틱스(simple heuristics)는 1995년 독일, 미국, 영국에서 심리학, 수학, 컴퓨터과학, 경제학, 생물학 등 학제간의 연구를 위해서 설립된 ABC(Adaptive Behavior and Cognition) 연구 그룹의 주된 이론이다 [3]. 이 연구소에서는 제한적 추리(bounded rationality)와 불확실성 하에서 좋은 의사 결정에 관한 연구를 해오고 있다. 간결한 휴리스틱스(simple heuristics)를 이용하려는 이유는 보안 관련 특징 변수와 관련된 사용자들의 선호도를 개량적으로 정확히 예측한다는 것이 쉬운 일이 아니기 때문이다. 추리(rationality)에는 두 종류가 있는데 무제한적 추리(unbounded rationality)와 제한적 추리이다. 무제한적 추리에 바탕을 두고 있는 의사 결정 모델에서는 실제로 사람이 겪고 있는 시간, 지식, 계산 능력 등의 제한이 거의 없다. 이 모델은 전통적으로 확률 이론에 바탕을 두었고, 기대 유틸리티의 최대화와 Bayesian 모델이 대표적인 것이다.

제한적 추리에 바탕을 두고 있는 의사 결정 모델에서는 만족화(satisficing)와 fast and frugal 휴리스틱스가 있다. 만족화는 가능한 경우의

선택을 찾아가고, fast and frugal 휴리스틱스는 여러 종류의 결정을 내리기 위해서 정보나 계산을 거의 이용하지 않는다. 그리고 이 휴리스틱스는 적응 가능한 선택을 위해서 최소한의 시간, 지식, 계산만을 필요로 한다. 의사 결정을 위해서 적은 시간과 지식을 필요로 한다는 것은 한 가지 이유에 의한 결정(one reason decision making)의 형태를 띠고 있다. 이 결정에서는 결정을 위해서 여러 가지 이유의 결합이 아니라 단지 하나의 정보만 필요하다는 것을 의미한다. 왜 한 가지 이유에 의한 결정이 타당성이 있느냐 하면, 첫째, 서로 다른 이유의 결합은 공통의 단위로 변경을 해야 하는데 이는 상당히 비용이 많이 드는 작업이다. 둘째, 모든 믿음, 소망 등에 대해서 개량적인 확률이나 유틸리티가 가능하다고 가정해도, 이것은 수학적으로 편리한 가정일 뿐이지 실세계는 그렇지 못한 경우가 많이 있다.

서로 다른 환경은 각기 특수한 fast and frugal 휴리스틱스를 필요로 한다. 그렇지만 약간의 차이점 때문에 모두 다른 휴리스틱스를 필요로 한다면 처리할 수 없을 정도의 많은 휴리스틱스를 만들어내어야 할 것이다. fast and frugal 휴리스틱스는 간결성 때문에 이런 어려움을 극복해주고 있고, 새로운 환경에서도 쉽게 일반화 할 수 있도록 해주고 있다. 간결한 휴리스틱스 중에서 대표적인 것은 Take The Best가 있는데, 이는 특징변수(cue)를 차례로 조사하여 두 개체를 차등화 시켜 줄 수 있는 변수를 발견하면 이 변수가 추론의 근거가 되고 나머지 특징변수는 모두 무시한다. Take The Best는 특히 훈련 집합(training set)의 규모가 작을 때 다중 회귀(multiple regression)를 능가한다[3].

2.3 SNMP(Simple Network Management Protocol)

SNMP (Simple Network Management Protocol)[4]는 1988년에 발표되었으며, 라우터, 서버, 워크스테이션, 그리고 다른 네트워크 자원들에 대하여 쉽게 구현할 수 있으며 오버헤드가 적은 네트워크 관리를 수행하기 위하여 설계되었다. SNMP는 NMS (Network Management System)에 설치된 SNMP Manager가 다른 네트워크 노드들에 설치된 SNMP Agent에게 “관리 정보”를 요청하는 클라이언트/서버 모델로 동작한다. 여기서 관리 정보는 SNMP MIB (Management Information Base)[5] 객체들이며 MIB의 표현방식과 구조에 대해서는 SMI(Structure of Management Information)[6]에서 정의하고 있다.

2.4 네트워크 보안 관리와 네트워크 관리

네트워크 보안 관리는 시간을 많이 소모하는 작업이며 많은 보안 전문가를 필요로 하는 일이다. 그래서 최근 통합 보안 관리(Integrated Security Management)에 대한 관심이 증가하고 있다. 이와 관련하여 각각의 보안 시스템에 서로 다른 보안 정책을 분배하기 위하여 정보 자산의 중요도를 고려하는 통합 보안 관리 시스템[7]과 이동 에이전트 기반의 네트워크 관리 시스템[8]이 제안된 바 있다.

정보 자산의 중요도를 고려하는 통합 보안 관리 시스템[7]의 경우에는 자

원의 중요도에 따라서 적용되는 보안 서비스를 달리함으로써 자원의 중요도를 고려하지 않는 경우에 비해서 시스템의 효율성을 증가시킬 수 있다는 장점을 가지고 있다. 그러나 단일변수에 의존하여 보안 정책을 결정함으로써 보안 시스템의 유연성이나 적응성이 떨어진다는 단점이 있다. 시스템의 보안을 고려함에 있어서 자원의 중요도뿐만 아니라 시스템에 가해지는 공격의 수준, 시스템에서 가용한 보안 서비스와 같은 다양한 변수들도 중요한 요소가 될 수 있다. 따라서 본 논문에서는 다중변수에 기반을 둔 적응적 보안 등급 알고리즘을 제안한다.

SNMP 기반의 네트워크 관리 모델과 이동 에이전트 기반의 네트워크 관리 모델을 비교하면, SNMP 기반의 중앙 집중형 네트워크 관리는 주기적인 폴링(polling)으로 인해 NMS(Network Management System)에게 관리 작업이 집중되므로 NMS의 처리 부하와 네트워크 트래픽이 증가한다는 문제점이 있다. 또한 이동 에이전트 기반의 모델은 수집된 데이터가 누적되는 양이 많아서 전송지연을 유발할 수 있다는 단점을 가진다. 따라서 혼합 모드가 제안되었는데 이 모델에서 이동 에이전트는 노드들로부터 수집된 데이터를 누적시키지 않고 관리되는 노드의 임시 저장소에 저장해 둔 채로 다른 노드로 이동하게 되고 관리되는 노드는 나중에 이동 에이전트와는 독립적으로 NMS에 데이터를 전송하게 된다[9].

그림 2의 (a)는 순수 SNMP 기반의 네트워크 관리 모델을 보여주고 있다. NMS에서 각 관리 노드(Managed Node)에 직접적으로 요청을 보내는 중앙 집중형 관리 모델이며 이는 네트워크가 혼잡할 때 네트워크 관리 자체를 위한 트래픽을 증가시켜 또 다른 혼잡을 야기할 수 있다는 문제점을 안고 있다.

그림 2의 (b)는 이동 에이전트 기반의 네트워크 관리 모델이다. 이 모델은 그림 2의 (a)의 순수 SNMP 기반 네트워크 관리 모델에 비하여 관리 트래픽의 집중은 피할 수 있지만 이동 에이전트가 각 관리 노드들을 이동함에 따라서 누적되는 관리 데이터가 증가하여 노드 간의 전송시간이 길어진다는 단점이 있다.

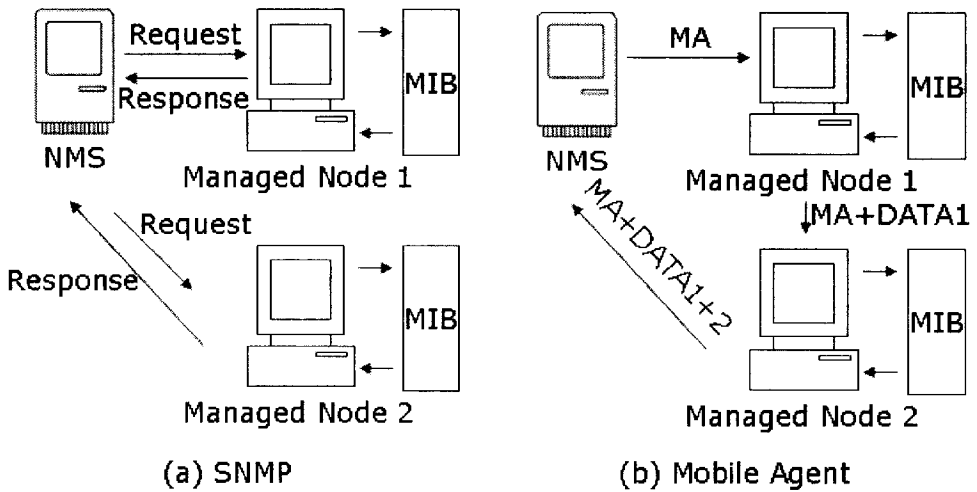


그림 2. SNMP 기반 모델과 이동 에이전트 기반 모델의 수행방식 비교

그림 3의 혼합 모드에서는 이동 에이전트가 관리 데이터를 누적시키지 않고 관리 노드의 임시 저장소에 저장한 후에 다음 관리 노드로 이동하여 관리를 수행한다. 그러면 관리 노드는 이동 에이전트의 동작과는 별개로 자신의 임시 저장소에 저장된 관리 데이터를 NMS에 전송한다. 이 모델은 그림 2의 (b)에 비하여 이동 에이전트가 관리노드들 사이를 이동할 때 관리 데이터를 누적시키지 않음으로써 이동 에이전트에 누적되는 데이터의 양을 줄이고 따라서 이동 에이전트가 다음 관리노드로 이동할 때 발생하는 전송 지연

을 줄일 수 있다. 또한 혼합모드를 사용함으로써 네트워크가 혼잡시 네트워크 관리 자체를 위한 트래픽을 증가시키고 이 트래픽이 NMS로 집중됨으로써 또 다른 혼잡을 야기할 수 있는 순수 SNMP 기반 관리 모델의 문제점을 해결할 수 있다.

따라서 본 논문에서는 네트워크 보안 관리 자체를 위한 네트워크 트래픽의 양을 줄이기 위하여 혼합 모드를 사용하는 네트워크 보안 관리 모델을 제안한다.

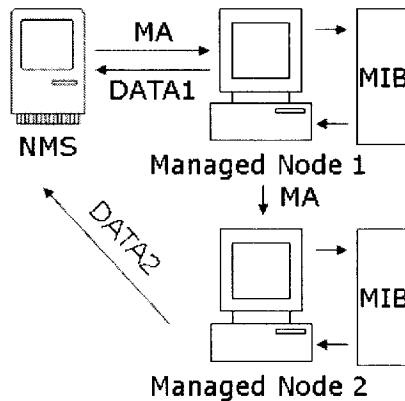


그림 3. 혼합 모드의 수행방식

2.5 기존 보안 모델의 정적인 정책 결정

기존의 보안 모델은 정책 결정에 정적인 의사결정 방법을 사용함으로써 환경의 변화에 적절히 대응하기 힘들며, 효율성이 부족하다. 예를 들면, 기존의 SSL/TLS는 Cipher Suite의 협상과정에서 클라이언트에서 전송된 Cipher Suite List에서 가능한 보안 강도가 높은 Cipher Suite를 선택한다

[16, 17]. 하지만 이러한 의사결정 방법은 자원이 열악한 사용자 단말에는 과부하와 지나치게 긴 대기시간을 초래할 수 있으며 암호 알고리즘은 대부분 컴퓨팅 자원을 많이 소모하므로 서버 측에도 과부하를 초래할 수 있다. 그림 4는 SSL/TLS Handshake 과정에 적응적 보안 모델을 적용하여 확장한 것이다[15]. Client Hello 단계에서 서버는 컨텍스트 엔진에 적응적 보안 등급을 요청하여 이에 따라 Cipher Suite를 선택한다. 환경이 변화하여도 기존의 SSL/TLS는 항상 가장 보안 강도가 높은 Cipher Suite를 선택하는데 비해 그림 4의 보안 모델[15]은 환경에 적절한 Cipher Suite를 선택할 수 있다.

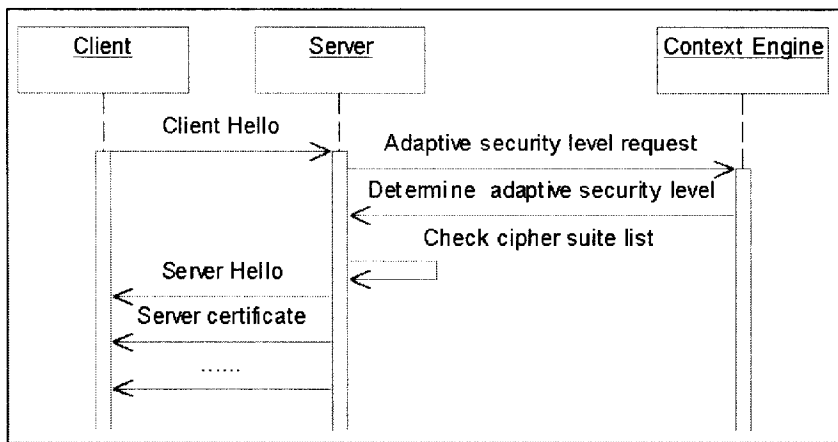


그림 4. 적응적 보안등급을 적용한 SSL/TLS Handshake

3. 적응적 네트워크 보안 관리 모델

이 장에서는, MAUT와 간결한 휴리스틱스를 사용하는 적응적 보안 등급 알고리즘을 소개하고 이 알고리즘과 SNMP, 이동 에이전트 기반의 적응적 네트워크 보안 관리 모델을 제안한다.

3.1 적응적 보안 등급 알고리즘

적응적 보안 등급 알고리즘이란 자원의 중요도와 같은 영역 의존 변수와 단말의 유형과 같은 영역 독립 변수에 따라서 동적으로 보안 등급을 적용하는 보안 정책 알고리즘이다.

이 알고리즘에서 사용되는 변수들은 다음과 같다.

1. 영역 독립 변수 $I = (i_1, i_2, \dots, i_n)$

네트워크 유형, 단말의 유형 등

보안 시스템을 사용하기 위한 기본적인 제약 조건으로 적용

2. 영역 의존 변수 $X = (x_1, x_2, \dots, x_n)$

시스템의 가용한 보안 시스템, 네트워크에 대한 공격 수준 등

적응적 정책 결정에 적용

3. 보안 등급 $SL(\text{Security Level}) = (0, 1, 2, \dots, 5)$

숫자가 증가할수록 보안 강도가 증가하며 이는 현재 네트워크의 보안 강도가 높다는 의미

보안 등급이 0인 경우에는 보안 시스템을 이용할 수 없음을 나타낸다.

다음은 적응적 보안 등급 알고리즘의 수학적 모델이다.

$$U = \sum_{i=1}^n k_i u_i(x_i), (0 \leq U \leq 1) \quad \text{수식 1. 전체 유틸리티 값의 계산}$$

$$SL = \lceil U * 10 \rceil / 2, (SL = 0, 1, \dots, 5) \quad \text{수식 2. 전체 유틸리티 값과 보안등급}$$

$$P_{SL} = \{p_0, p_1, \dots, p_5\} \quad \text{수식 3 보안 서비스의 전체 속성들의 집합}$$

$$p_i = \{A_i, R_m, SC_o, M_q, AC_s, H_i\} \quad \text{수식 4 각 보안 속성들}$$

u_i 는 다양한 환경변수(x_i)들을 유틸리티 값으로 변환한 값이며, k_i 는 변수들의 가중치로서 보안정책과 사용자의 보안선호도에 따라서 결정된다. U 는 전체 유틸리티 값이다. SL 은 0에서 5까지의 수로 보안등급을 나타내며 값이 0인 경우는 보안시스템을 사용할 수 없는 경우이다. P_{SL} 는 보안시스템의 속성인 p_i 들의 집합이며 관리자에 의해서 결정된다. A_i 는 인증기법, R_m 는 프로토콜, SC_o 는 대칭 암호 알고리즘, M_q 는 운영모드, AC_s 는 공개키 암호 알고리즘, H_i 는 해쉬 알고리즘의 종류를 나타낸다.

다양한 변수들에 의해서 SL 이 결정되면 SL 에 따라서 p_i 를 선택하여 보안 서비스를 제공하게 된다.

표 1. 인증기법과 프로토콜

A_I	인증기법	R_m	프로토콜
A_0	OTP ¹⁾ 기반	R_0	SPKI ²⁾
A_I	부가형 인증 ³⁾	R_I	Wireless PKI
		R_2	PKI

표 2. 대칭 암호 알고리즘과 운영모드

SC_j	대칭 암호 알고리즘-키길이	M_m	운영모드 ⁴⁾
SC_0	DES	M_0	ECB
SC_1	3DES	M_1	OFB
SC_2	AES-128	M_2	CFB
SC_3	AES-192	M_3	CBC
		M_4	CTR

표 3. 공개키 암호 알고리즘과 해쉬 알고리즘

AC_i	공개키 암호 알고리즘-키길이	H_m	해쉬 알고리즘
AC_0	RSA-512	H_0	MD5
AC_1	RSA-1024	H_1	SHA-1
AC_2	RSA-2048		

1) OTP: One Time Password[10, 11]

2) SPKI(Simplified Public-Key Infrastructure)[12, 13, 19]는 CA를 기반으로 하는 기존의 PKI의 CRL 관리의 복잡함을 줄이기 위하여 PKI를 간소화하였다.

3) 부가형 인증은 OTP에 전자서명이나 생체인식과 같은 다른 인증기법들을 부가한 인증방법을 말한다.

4) 대칭 암호 알고리즘의 운영모드는 각각 다른 용도를 가지므로 서로간의 우위비 교와는 관계없이 배열되었다. ECB 모드: Electronic CodeBook 모드, OFB 모드: Output FeedBack 모드, CFB 모드: Cipher FeedBack 모드, CBC 모드: Cipher Block Chaining 모드, CTR 모드: Counter 모드

적응적 보안 등급을 결정하기 위한 전체 알고리즘[14]은 다음과 같다.

표 4. SecurityLevel(*securityProblem*) 함수

```
SecurityLevel(securityProblem)  
// securityProblem: 보안 등급 결정 문제  
// 영역 독립 변수 이용  
  calculate SL by I end  
  if SL = 0 then  
    return SL // 보안 시스템을 사용할 수 없음  
  
// 영역 의존 변수 이용(MAUT와 간결한 휴리스틱스 중에서 선택)  
  if MAUT then SL = MAUT(X)  
  if Simple Heuristics then SL = TakeTheBest(X);  
  return SL;  
end;
```

표 5. MAUT(*X*) 함수

```
MAUT(X)  
// MAUT에 따른 사용자와의 상호 작용에 의해서 전체 유틸리티 함수 결정  
 $u(x_1, x_2, \dots, x_n) = k_1 u_1(x_1) + k_2 u_2(x_2) + \dots + k_n u_n(x_n)$   
//  $k_i$ : 전체 영역 의존 변수 X에 대한 조정 상수  
//  $x_i$ : 영역 의존 변수,  $u_i(x_i^0) = 0$ ,  $u_i(x_i^*) = 1$   
  
  ask the user's preference and decide  $k_i$   
  for i = 1 to n  
    do  $u_i(x_i) = \text{GetUtilFunction}(x_i)$ ;  
  end  
  return  $u(x_1, x_2, \dots, x_n)$ ;  
end;
```

표 6. TakeTheBest($u(x_1, x_2, \dots, x_n)$) 함수

```

TakeTheBest( $u(x_1, x_2, \dots, x_n)$ )
//  $x_i$ 에 대하여 가장 선호되는 변수를 찾고 나머지는 무시
 $u(x_1, x_2, \dots, x_n)$  is calculated by only considering  $x_i$ ;
SL is calculated by the value of  $u(x_1, x_2, \dots, x_n)$ ;
return SL;
end;

```

표 7. GetUtilFunction(x_i) 함수

```

GetUtilFunction( $x_i$ )
// 사용자의 선호도에 따라서 유틸리티 함수를 결정
//  $x_i$  : 영역 의존 변수
uRiskProne : 사용자가  $x_i$ 에 대하여 위험 선호 - convex
uRiskNeutral : 사용자가  $x_i$ 에 대하여 위험 중립 - linear
uRiskAverse : 사용자가  $x_i$ 에 대하여 위험 회피 - concave
 $x$  :  $x_i$ 로부터 선택된 임의의 값
 $h$  : 임의로 선택된 값
 $\langle x+h, x-h \rangle$  :  $x+h$ 에서  $x-h$ 까지의 제비뽑기
// where the lottery ( $x^*, p, x^o$ ) yields a  $p$  chance at  $x^*$ 
// and a  $(1-p)$  chance at  $x^o$ 

ask user to prefer  $\langle x+h, x-h \rangle$  or  $x$  // 사용자와의 상호 작용
if user prefer  $\langle x+h, x-h \rangle$  then
    return uRiskProne; // 예를 들면,  $u = b(2^x - 1)$ 
elseif user prefer  $x$  then
    return uRiskAverse; // 예를 들면,  $u = \log_2(x+1)$ 
else
    return uRiskNeutral; // 예를 들면,  $u = bx$ 
end;

```

3.2 보안 정책과 접근 정책

보안 정책은 적응적 보안 등급 알고리즘에 따라서 보안 등급을 결정하며 접근 정책은 보안 등급과 컨텍스트 정보에 따라서 사용자의 요청을 허가 또는 거부한다. 표 8은 보안 정책의 예이다. 정책에 적용되는 변수들은 다음과 같다.

- x_{sec} : 시스템에서 가용한 보안 서비스의 보안 강도
- x_{att} : 두 시스템 사이의 네트워크에 대한 공격자의 공격 정도
- x_{def} : 두 시스템 사이의 네트워크의 보호 수준
- x_{sen} : 사용자가 접근하려고 하는 자원의 중요도를 나타낸다.
- x_{pre} : 사용자의 보안 선호도
- nType : 네트워크 유형
- tType : 단말의 종류

표 8. 보안 정책의 예

보호되는 자원 A에 대한 보안 정책	
동작	읽기, 쓰기
유틸리티 함수	$u(x_{sec}, x_{att}, x_{def}, x_{sen}, x_{pre}) = k_{sec}u(x_{sec}) + k_{att}u(x_{att}) + k_{def}u(x_{def}) + k_{sen}u(x_{sen}) + k_{pre}u(x_{pre})$
제약 조건	$nType = 100 \text{ Kbps}; tType = \text{PC/PDA/Cell};$
유틸리티 함수	$uRiskProne = 2^{2(x-1)}$ $uRiskNeutral = x$ $uRiskAverse = \log_2(x+1);$

보안정책은 적응적 보안등급 알고리즘에 의해서 보안등급을 결정하고 적용할 보안 서비스의 속성들을 결정한다. 표 9의 환산표는 변수들을 정량적인 유틸리티 값으로 변환하는데 사용된다.

표 9. 환경 변수의 환산표

	0.25	0.50	1.0
x_{att} (공격 수준)	없음	보통	심각
x_{def} (보호 수준)	보호되지 않음	보통	강하게 보호됨
x_{sec} (가용 보안 서비스)	사용할 수 없음	안전함	매우 안전함
x_{sen} (자원의 중요도)	공개자원	기밀자원	최고 기밀 자원
x_{res} (단말의 연산능력)	열악함	보통	성능이 좋음

표 10. 접근 정책의 예

보호되는 자원 A에 대한 접근 정책
If $SL \geq 2$ and (Role = administrator or Role = user and Date = Weekdays and Time = daytime) Then resource A can be read
If $SL \geq 3$ and Role = administrator Then resource A can be written

표 10은 접근 정책의 예이다. 보안등급, 사용자의 역할, 그리고 시간에 의해서 위 정책에서는 읽기 또는 쓰기 권한이 할당된다. SL 은 접근 가능한 보안 등급의 하한값을 뜻하며, 2보다 보안 등급이 낮은 경우에는 어떠한 사용자도 읽기 권한을 가질 수 없음을 뜻하며, 만약 사용자의 역할이 관리자이고 보안 등급이 3보다 높으면 사용자는 쓰기 권한을 획득할 수 있다.

3.3 ASMIB(Adaptive Security MIB)

본 논문에서는 SNMP의 MIB를 확장하여 ASMIB를 정의하는데 ASMIB는 다섯 유형의 그룹으로 구성된다.

- **SystemInfo** 그룹: 시스템과 관련된 정보. 호스트 이름, 시스템의 종류, CPU의 연산 능력, 메모리의 크기, CPU 사용율, 메모리 사용율 등
 - **NetworkInfo** 그룹: 네트워크와 관련된 정보. IP 주소, 네트워크 유형, 네트워크의 속도, 수립된 연결의 수, 네트워크 사용율 등
 - **AdaptiveSecurity** 그룹: 가용 보안 서비스, 자원의 보호 수준, 네트워크의 보호 수준과 같은 영역 의존 변수와 적응적 보안 등급 알고리즘에 의해서 결정되는 적응적 보안 등급 등
- Policy** 그룹: 적응적 보안 등급을 결정하기 위한 보안 정책과 자원에 대한 접근을 허가하는 접근 정책

3.4 전체 시스템 구조

중앙집중형 보안 관리 시스템의 트래픽 혼잡시 관리 트래픽이 과도하게 증가하는 문제점을 피하기 위하여 본 논문에서는 이동 에이전트와 SNMP에 기반한 분산 구조의 모델을 제안한다.

그림 6은 본 모델의 두 가지 주요 구성요소를 나타내고 있다. Context Engine은 Security Module, Adaptive Security Level Algorithm, MA(Mobile Agent) Generator and SNMP Manager로 구성된다.

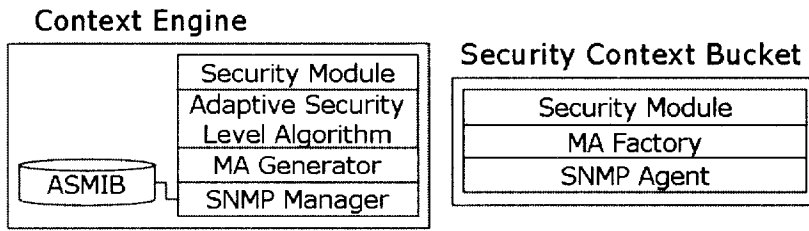


그림 5. Context Engine과 Security Context Bucket

그림 5의 Context Engine은 NSMS(Network Security Management System)에 설치되어 Security Module을 사용하여 기본적인 보안 서비스를 제공하며 적응적 보안 등급 알고리즘(Adaptive Security Level Algorithm)에 의해서 보안 등급을 결정하고 보안 서비스에 적용될 암호학적 매개변수들을 결정하는 역할을 수행하며 MA Generator를 사용하여 이동 에이전트를 생성한다. 그리고 SNMP Manager를 사용하여 SNMP Agent에 의해서 전송되는 관리 정보인 ASMIB를 수집한다.

그림 5의 Security Context Bucket은 관리 노드에 설치되어 NSMS에 ASMIB를 제공하는 역할을 수행한다. MA Factory는 이동 에이전트들이 이동하는 영역을 제공하며, Security Module은 이동 에이전트들을 인증한다.

Context Engine으로부터 이동 에이전트가 생성되면 이동 에이전트는 계획된 여러 노드들을 이동하면서 ASMIB 정보를 수집하고 Security Context Bucket의 임시저장소에 저장한다. 그러면 Security Context Bucket은 이동 에이전트와는 별개로 SNMP의 비동기 전송 방식인 Trap을 사용하여 저장된 ASMIB 객체를 전송한다. 적응적 보안 등급 알고리즘은 수집된 ASMIB에 의해서 보안 등급과 보안 서비스를 위한 속성들을 결정한다. 그리고 Context Engine의 Security Module은 보안 등급과 이러한 속성들에 따라서

보안 서비스를 제공하게 된다.

그림 6의 혼합 모드는 이동 에이전트와 SNMP Trap을 사용함으로써 SNMP 기반의 요청과 응답으로 이루어진 폴링(polling)을 사용할 때 발생할 수 있는 네트워크 자원의 과도한 사용을 줄일 수 있다. 이동 에이전트는 시스템 내부의 변수들 이외에 사용자의 보안 선호도와 같은 영역 의존 변수들을 획득할 수 있는 능력을 갖추기 위해서 적응적 보안 등급 알고리즘을 탑재한다.

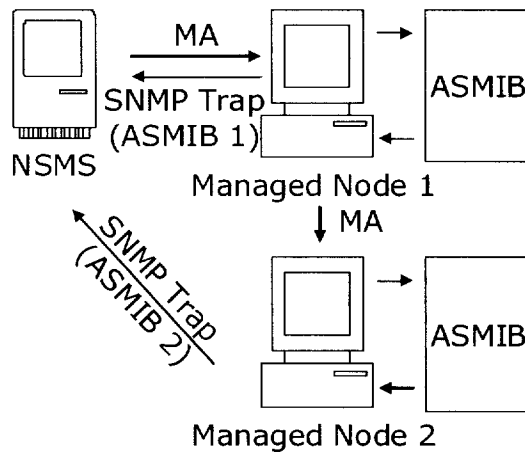


그림 6. SNMP Trap을 사용한 혼합모드

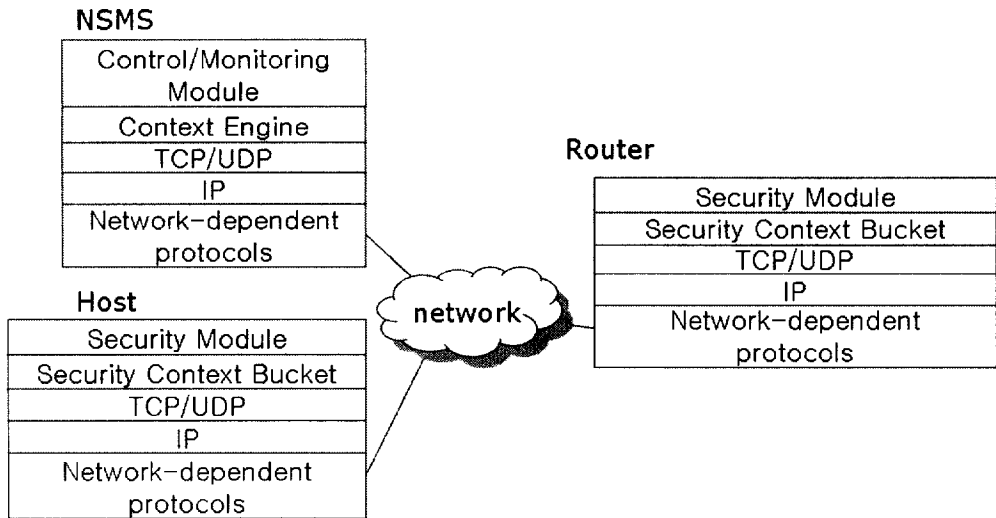


그림 7. 전체 시스템 구조의 예

그림 7은 본 논문에서 제안된 전체 시스템 구조를 나타내고 있다. NSMS는 Context Engine의 상위에 Control/Monitoring Module을 가지며 관리자는 이 모듈을 통하여 네트워크 자원에 대한 보안 관리를 수행하게 된다. 라우터, 호스트와 같은 각 관리 노드들에 설치되는 Security Context Bucket은 NSMS에 관리 정보를 제공한다.

NSMS(Network Security Management System)의 역할은 다음과 같다.

- 이동 에이전트를 생성하여 ASMIB를 수집하고, 관리되는 노드들로부터 Trap PDU(Protocol Data Unit)을 받는다.
- 보안/서비스 정책들을 수립하고 관리되는 노드들에 이를 적용한다.

3.5 Proxy에 의한 NSMS의 계층적 분산

제안된 모델은 폴링 기반의 ASMIB의 수집 대신에 이동 에이전트와 Trap 프로토콜에 기반한 수집을 함으로써 관리 정보를 전송하는 트랜잭션의 수를 줄일 수 있다. 그러나 보안 정책과 서비스 정책들에 대한 처리 과정은 여전히 집중되어 있다. 따라서 Proxy에 의한 NSMS의 계층적 분산을 소개한다.

여기서 NSMS의 역할은 다음과 같이 확장된다.

보안 정책과 서비스 정책을 Proxy에 배포한다.

Proxy의 역할은 다음과 같다.

- 이동 에이전트를 통하여 ASMIB를 수집하고 관리되는 노드들로부터 Trap PDU를 받는다.
- 수집된 ASMIB를 Trap을 사용하여 NSMS에 전달한다.
- 관리되는 노드들에 보안 정책과 서비스 정책을 적용한다.
- SNMP를 사용하지 않는 노드들을 관리한다.

그림 8는 Proxy의 시스템 구성을 나타내고 있다. NSMS에 대해서는 관리 노드처럼 ASMIB를 제공하기 위해서 Security Context Bucket을 포함하고 있으며, 관리 노드에 대해서는 NSMS처럼 관리 기능을 수행하기 위해서 Context Engine을 포함한다.

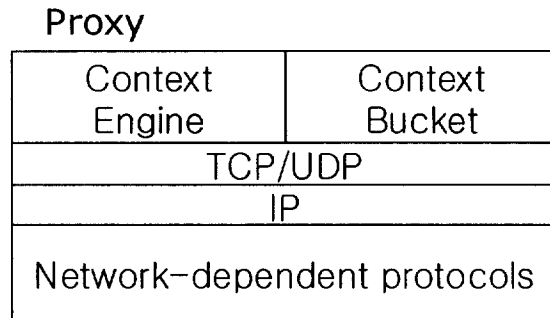


그림 8. Proxy

그림 9은 Proxy에 의한 NSMS의 계층적 분산을 나타내고 있다. 각 관리 도메인은 NSMS의 입장에서는 하나의 관리 노드로써 동작하게 되고 각 Proxy는 하부의 관리 노드들을 관리하는 계층적인 분산 관리 모델을 나타내고 있다.

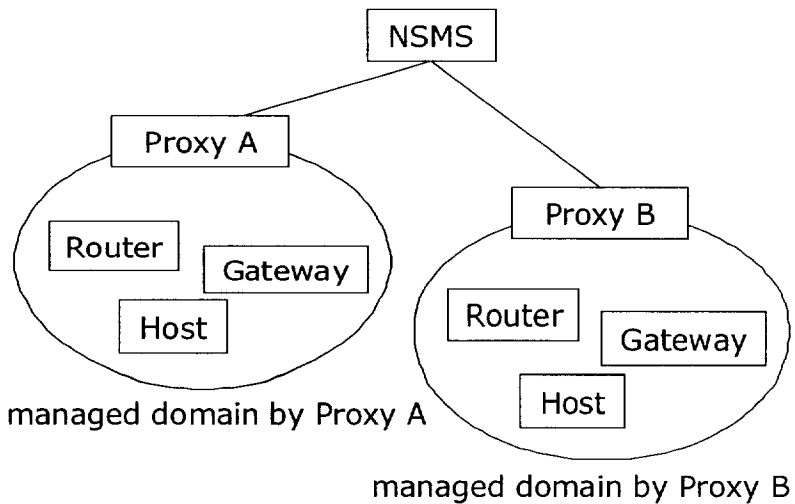


그림 9. Proxy에 의한 NSMS의 계층적 분산

3.6 적응적 보안 등급의 동적인 적용

사용자 또는 네트워크의 환경이 변화하여 보안등급이 변화하게 되면 이것을 현재 사용 중인 보안 서비스에 동적으로 반영할 수 있어야 한다. 이는 보안 등급에 따라 정해진 표 7, 8, 9과 같은 보안 서비스의 속성들을 동적으로 변화시킴으로써 가능하다.

3.6.1 인증 기법의 동적인 적용

그림 10은 해쉬 함수의 일방향성에 의하여 인증기법을 제공하는 OTP(One-Time Password)[10, 11] 기반의 인증과정을 나타낸다. OTP 기반 인증은 대칭 암호화나 전자서명을 사용하지 않으며 해쉬 함수의 일방향성에 의한 단순한 인증 기법을 제공한다.

OTP 기반의 인증 시스템의 특징은 다음과 같다.

사용하기에 간단하다.

비밀 패스워드(pass_phrase)를 기억하도록 한다.

- 자동화가 될 수 있다.
- 알고리즘이 공개되어 있다. (MD4, MD5, SHA1 해쉬 알고리즘)
- 어떠한 비밀 정보도 호스트에 보관되지 않는다.

해쉬 알고리즘은 $f(x) = y$ 에서 x 를 알면 y 를 쉽게 구할 수 있으나 y 를 알고 x 를 알아내는 것이 계산상 불가능하다는 특징을 가진다. OTP 기반의 인

증 시스템은 이러한 해쉬 알고리즘의 일방향성을 이용하여 다음과 같은 인증 과정을 수행한다.

첫 번째 인증값 $P_{(1)}$ 은 사용자의 비밀 패스워드(S)를 인증 서버에서 요청한 n 만큼 해쉬 알고리즘 $H(x)$ 를 반복해서 적용함으로써 생성된다.

$$S = \text{seed} \parallel \text{password}$$

$$P_{(1)} = H_n(S)$$

다음 번 인증 과정에서는 $n-1$ 반복하여 해쉬 알고리즘을 적용함으로써 생성되는 인증값 $P_{(2)}$ 를 사용한다.

$$P_{(2)} = H_{n-1}(S)$$

이러한 과정을 거치기 때문에 공격자가 어느 순간의 인증값 $P_{(i)}$ 를 알아냈다고 해도 다음번 인증값 $P_{(i+1)}$ 을 알아내는 것이 계산상 불가능하다.

표 1은 OTP 기반 인증과 OTP 기반 인증에 전자서명 또는 생체인식을 부가한 형태의 인증기법들을 기술하였다. OTP 기반 인증에 전자서명을 부가하는 경우에는 아래 수식 5, 6과 같이 OTP의 서버로 전송될 output(Computation Step의 결과로 생성된 인증값)에 OTP 클라이언트의 전자서명을 추가한다. Sig(x)는 x에 대하여 전자서명을 생성한다는 의미이다. 그리고 생체인식을 부가하는 경우에는 OTP에 의해 생성되는 인증값 $H_n(S)$ 에 생체인식 데이터를 접합하여 인증 과정을 진행하게 된다. OTP 인증 과정에 다른 기법들을 부가함으로써 기존의 OTP 기반의 인증 과정을 크게 변경하지 않고도 보다 강력한 인증 기법을 제공할 수 있는 장점을 가지고 있다.

$P_{(1)} = \text{Sig}(H_n(S))$ 수식 5 OTP + 전자서명(Computation Step)

$P_{(2)} = \text{Sig}(H_{n-1}(S))$ 수식 6 OTP + 전자서명(Computation Step)

$P_{(1)} = H_n(S) \parallel (\text{인식데이터})$ 수식 7 OTP + 생체인식 (Computation Step)

$P_{(2)} = H_{n-1}(S) \parallel (\text{인식데이터})$ 수식 8 OTP + 생체인식 (Computation Step)

환경의 변화에 의해 보안 등급이 변경되어 보다 강력한 인증 기법이 요구되는 경우에 위와 같이 전자서명을 부가하거나 생체인식을 부가함으로써 환경의 변화에 대응할 수 있다.

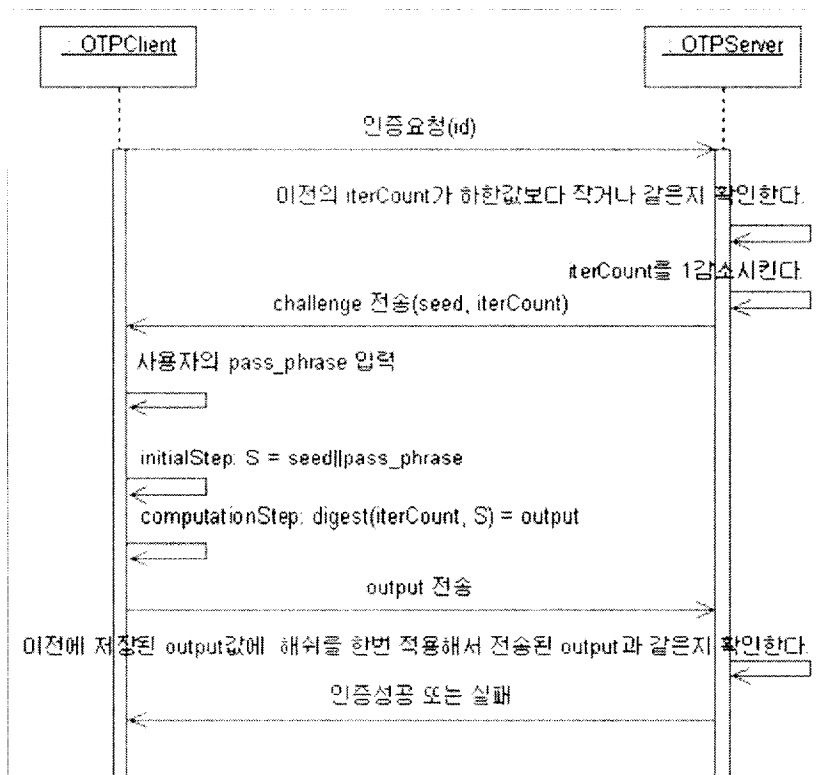


그림 10. One Time Password System의 인증 과정

3.6.2 대칭 암호화 기법의 동적인 적용

표 2는 보안 등급의 변화에 따른 대칭 암호 알고리즘의 변화에 대해서 나타내고 있다. 환경의 변화에 의해서 비밀성에 대한 보안 요구가 증가하게 되면 보다 강한 대칭 암호 알고리즘을 적용할 필요가 있다. 이 경우에 알고리즘이 변경되면 사용할 비밀키도 항상 달라야 한다. 왜냐하면, 이전의 대칭 암호 알고리즘 E_{n-1} 과 변경된 대칭 암호 알고리즘 E_n 에 대해서 동일한 비밀키 K_{n-1} 을 적용하는 경우에 E_{n-1} 의 암호학적 비도가 E_n 보다 낮다고 가정하면 공격자는 E_n 보다는 E_{n-1} 에 대해서 공격을 수행하여 K_{n-1} 를 획득함으로써 암호학적 비도가 높은 E_n 에 대한 공격을 수행할 수 있기 때문이다. 따라서 다음과 같은 키 생성과정을 따라서 비밀키 K_n 을 생성한다.

먼저, 통신 당사자 사이의 공유 비밀값인 MS 를 Diffie-Hellman 키 일치 알고리즘을 이용하여 생성한다. 그리고 통신당사자 사이의 공유정보인 난수 값, $Seed_n$ 과 공개정보인 해쉬함수 $H(n)$ 을 사용하여 아래와 같이 비밀키를 생성한다.

표 11. 비밀키 생성과정

공개정보: $i, H(x)$ 통신 당사자 사이의 공유정보: $Seed_n$ 비밀정보: MS, K_n 키 생성 과정 $K_{n-1} = H_i(MS \ Seed_{n-1})$ $K_n = H_i(MS \ Seed_n)$

$Seed_n$ 은 한 번의 키 생성에 대해서만 사용되는 난수값으로써 반복해서 사용되어서는 안 되며 n 은 몇 번째 비밀키 생성인지 나타낸다. i 는 해쉬함수의 반복횟수를 나타낸다. 여기서, MS 는 통신 당사자들 사이의 공유 비밀값으로써 절대 공개되어서는 안 된다.

위와 같은 키 생성과정을 거침으로써 대칭 암호 알고리즘을 변경하더라도 각각의 대칭 암호 알고리즘을 위한 비밀키를 안전하게 생성할 수 있을 것이다.

3.7 적응적 네트워크 보안 관리 모델의 보안 관리 절차

그림 11은 본 모델이 네트워크 보안 관리를 수행하는 절차에 대해서 나타낸 그림이다. 여기서 Proxy와 NSMS는 MA(이동 에이전트)를 사용하여 관리정보를 수집한다. 이때, 혼합모드를 사용한다. 그런데 MA를 생성하는 주기에 대해서는 보다 연구가 필요하다. 만약 주기가 너무 짧아진다면 결국 SNMP 기반 보안 관리 모델과 유사해지게 되고, 주기가 너무 길어진다면 환경변수들의 변화를 적절히 감지할 수 없을 것이다.

보호된 자원에 대한 접근 요청이 있으면 보안 정책과 접근 정책에 의해서 요청을 허가할 것인지 거부할 것인지 결정한다. 그리고 자원에 대한 접근도중에 MA에 의해서 환경 변수의 변화가 확인되면 이에 따라서 보안 서비스의 속성들을 변화시켜서 보안 서비스를 제공하게 된다.

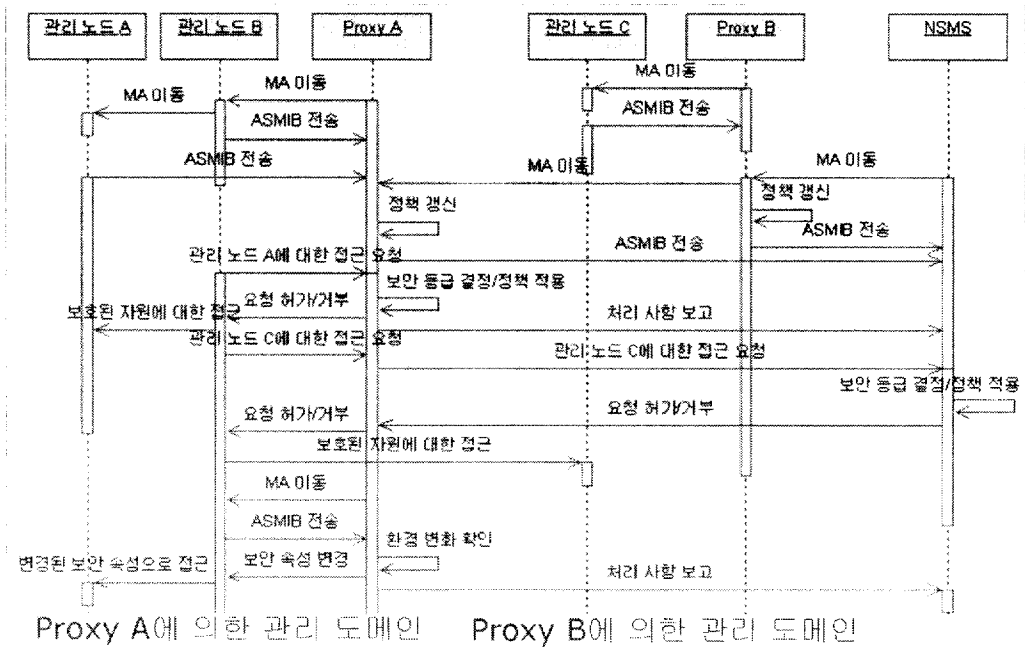


그림 11. 적응적 네트워크 보안 관리 모델의 보안 관리 절차

4. 구현 및 평가

본 논문에서 제안하고 있는 적응적 네트워크 보안 관리 시스템의 효율성을 평가하기 위하여 먼저, 시스템의 핵심을 이루고 있는 적응적 보안 등급 알고리즘과 보안 모듈에 대하여 구현한다. 그리고 이를 확장하여 SNMP와 이동 에이전트 기반 적응적 네트워크 보안 관리 시스템을 구현한다.

4.1 적응적 보안 등급 알고리즘과 보안 모듈

적응적 보안 등급 알고리즘과 보안 서비스의 동적인 적용과정의 효율성을 평가하기 위하여 다음과 같은 시스템을 구현한다.

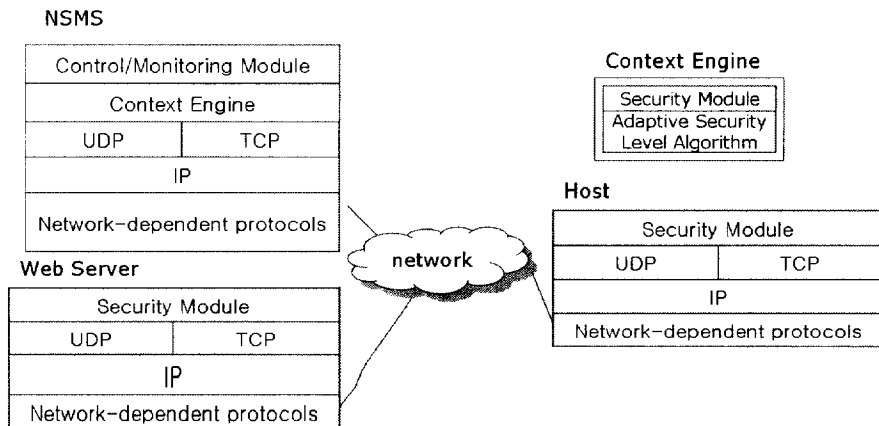


그림 12. 구현 시스템

여기서 Context Engine은 적응적 보안 등급 알고리즘을 탑재하고 있으며 Security Module은 보안 서비스를 동적으로 적용하는 역할을 수행한다. 위 시스템은 그림 8 전체 시스템 구조에서 나타난 시스템에 비하여 적응적 보

안 등급 알고리즘에 대한 평가의 정확도를 높이기 위하여 구성을 간소화하였다.

① 구현 환경

NSMS

- Language : JDK 1.4.2, Servlet 2.3, XML
- 보안 API : Bouncycastle

Web Server

- Language : JDK 1.4.2, Servlet 2.3, XML
- Web Server : Tomcat 4.01
- 보안 API : Bouncycastle

Host

- Language : JDK 1.4.2, XML
- 보안 API : Bouncycastle

4.1.1 평가

위 시스템에 적용되는 보안 정책과 서비스 정책은 다음과 같다.

표 12. 보안 정책

보호되는 자원에 대한 보안 정책	
동작	읽기, 쓰기
유틸리티 함수	$u(x_{conn}, x_{pro}, x_{res}, x_{role}) = k_{conn} u(x_{conn}) + k_{pro} u(x_{pro}) + k_{res} u(x_{res}) + k_{role} u(x_{role})$
제약 조건	$tType = PC/PDA/Cell;$
사용자의 선호도	$uRiskProne = 2^{2(x-1)}$ $uRiskNeutral = x$ $uRiskAverse = \log_2(x+1);$

x_{conn} 은 현재 생성된 네트워크 연결의 수를 나타낸다. 연결의 수가 증가할수록 환산된 값은 증가하여 보안 등급을 높게 된다. x_{pro} 는 웹 서버와 클라이언트 사이의 네트워크의 보호 수준을 나타낸다. 보호 수준이 낮을수록 환산된 값은 증가한다. 이는 보호 수준이 높으면 추가적인 보안을 제공할 필요가 적음을 나타낸다. x_{res} 는 접근하려는 자원의 중요도이다. 중요도가 높은 자원에 접근하려 할수록 환산된 값은 증가하게 된다. x_{role} 은 자원에 대한 접근 요청의 권한을 나타낸다. 보다 중요한 권한일수록 환산된 값은 증가하게 된다. 따라서 현재 생성된 네트워크의 연결의 수가 많을수록, 웹 서버와 클라이언트 사이의 네트워크의 보호 수준이 낮을수록, 자원의 중요도가 높을수록, 그리고 요청의 권한이 클수록 보안 등급은 높아지게 된다.

표 13. 환경 변수의 환산표

	0.25	0.5	0.75	1.0
x_{conn} (연결 수)	10	50	100	100이상
x_{pro} (보호 수준)	강하게 보호됨	보호됨	약하게 보호됨	보호되지 않음
x_{res} (자원의 중요도)	공개가능	보통	중요	매우 중요
x_{role} (권한)	Guest	일반 사용자	관리자	시스템

이렇게 보안 등급이 결정되면 다음과 같은 접근 정책이 적용된다.

표 14. 접근 정책

접근 정책
If SL >= 4 and (Role = 관리자 or Role = 인증된 사용자) Then resource A can be read or written
If SL >= 2 and Role = 일반 사용자 Then resource A can be read

위의 첫 번째 접근 정책은 보안 등급이 4이상이고 관리자 또는 기존의 인증된 사용자에게 대해서는 접근을 허가한다. 이는 보안 등급이 상승할 경우에 기존의 인증된 사용자들에게는 계속 서비스를 제공하지만 새로 접근하는 사용자는 관리자 이외에는 접근이 불가함을 나타내고 있다. 이는 네트워크 혼잡이나 공격자의 공격에 의해서 보안 등급이 상승하더라도 기존의 사용자는 계속 서비스할 수 있도록 하여 가용성을 높이기 위한 것이다.

여기서 NSMS의 관리자가 정책을 결정할 때 적응적 보안 등급 알고리즘에 의해서 각 변수에 대한 가중치와 유틸리티 함수의 유형을 결정한다. 이때, 관리자가 시스템의 가용성을 중시한다면, x_{comm} (연결 수)에 대한 가중치가 증가할 것이다. 마찬가지로 시스템의 비밀성이나 무결성을 중시한다면, x_{pro} (보호 수준)에 대한 가중치가 증가할 것이다.

표 15. 인증기법과 해쉬 알고리즘

A_l	인증기법	H_m	해쉬 알고리즘
A_0	OTP 기반	H_0	MD5
		H_1	SHA
		H_2	SHA256

표 16. 대칭 암호 알고리즘과 운영모드

SC_j	대칭 암호 알고리즘 키길이	M_m	운영모드
SC_0	DES	M_0	CBC
SC_1	3DES	M_1	CTR
SC_2	AES-128		
SC_3	AES-192		

그림 12의 구현 시스템에서 보안 서비스의 속성들을 변화시킬 때의 연산량의 변화에 대해서 분석하기 위하여 인증기법들과 대칭암호알고리즘의 평균 응답시간에 대해서 다음과 같이 측정하였다.

표 17. 키 생성과 대칭암호의 응답시간 비교 (단위: millisecond)

	DH	키 생성	DES	3DES	AESLight ⁵⁾	AESFast
평균응답시간	231	5	5	10	4	2

표 17에서 공유 비밀값인 MS를 생성하는데 1024비트 Diffie-Hellman 알고리즘을 사용하였으며 각 단계의 비밀키를 생성하기 위해 SHA1을 1000번 반복하여 사용하였으며 대칭암호는 10Kbyte에 대해서 암호화/복호화하였다.

표 17은 공유 비밀값을 생성하기 위해서는 시간이 많이 소요되지만 한 번 공유 비밀값을 생성하고 나면 비밀키를 생성하기 위한 응답시간이 다른 대칭암호 연산의 응답시간에 비해서 그다지 크지 않음을 알 수 있다. 특히, 키 생성에 필요한 시간은 고정되어 있지만 대칭암호 연산은 평문의 크기가 증가할수록 응답시간이 증가한다. 이는 평문의 크기가 충분히 클 경우에는 비밀키의 생성에 걸리는 시간이 전체 암호화에 걸리는 시간에 크게 영향을 미치지 않는다는 의미이다. 따라서 대칭암호알고리즘을 변화시키기 위한 연산량의 증가는 비밀성을 제공하는데 있어서 크게 영향을 미치지 않음을 알 수 있다.

5) AESLight와 AESFast는 Bouncycastle의 AESLightEngine과 AESFastEngine을 사용하였음을 의미한다[20].

표 18. 인증기법들의 응답 시간 비교

	인증데이터 크기(byte)	평균 응답시간(millisecond)
OTP 기반	346	45
OTP+전자서명	371	67
비고	XML 태그 포함, SHA1, RSA1024	인증서 검증시간 제외

표 18은 OTP 기반 인증과 OTP 기반 인증에 전자서명을 부가한 인증 기법의 응답 시간과 데이터 크기를 비교하고 있다. 전자서명을 부가했을 때 약 48%의 응답 시간의 증가가 관찰되었다.

4.2 SNMP와 이동 에이전트 기반 적응적 네트워크 보안관리 시스템

① 구현 환경

Language : JDK 1.4.2, J2ME, Servlet

- 보안 API : Bouncycastle 1.21

- SNMP API : Java Dynamic Management Kit 5.0

이동 에이전트 API : Aglets Software Development Kit 4 beta

4.1의 구현 시스템을 바탕으로 하여 그림 8의 SNMP와 모바일 에이전트 기반 시스템을 구현한다.

4.2.1 평가

적응적 보안 등급 알고리즘은 사용자와 시스템 사이의 많은 상호 작용을

요구한다는 점 때문에 다소 복잡하다.

제안된 알고리즘에서 사용자가 MAUT와 조정 상수 $K = \{k_1, k_2, \dots, k_i\}$ 를 선택했다고 가정하고 $X = \{x_1, x_2, \dots, x_i\}$ 는 영역 의존 변수의 집합이라고 가정하자. I 는 적응적 보안 등급을 결정하는데 요구되는 상호작용의 수를 나타낸다고 하면 I 는 수식 9에 의해서 다음과 같이 표현된다.

$$I = (|K| - 1) + |X| \approx 2|X|, \text{ 변수들은 덧셈 독립이며 유틸리티 독립 수식 9}$$

사용자의 선호도에 따라서 조정 상수를 결정하기 위하여 적어도 $(|K| - 1)$ 번의 상호 작용이 요구된다. 세 가지의 변수의 경우에, 알고리즘은 k_1 과 $(k_2 + k_3)$ 에 대한 선호도를 질의한다. 그리고 k_2 와 k_3 에 대한 선호도를 질의한다. 이러한 과정을 거쳐서, 조정 상수들이 결정된다. 결국, $(|K| - 1)$ 은 조정 상수들을 결정하기 위한 최소한의 상호 작용의 수이다. 또한 사용자의 위험 선호에 따라서 유틸리티 함수를 결정하기 위하여 최소한 $|X|$ 번의 상호 작용이 필요하다. 그리고 변수들이 덧셈 독립이고 유틸리티 독립이라는 가정에 의해서 조정 상수의 수는 영역 의존 변수들의 수와 같다.

그런데, 이러한 상호 작용들은 요청/응답 방식에 의해서 이루어지므로 실제 NSMS와 관리노드 사이의 네트워크 전송의 수인 T 는 수식 10에서 보듯이 수식 9의 I 의 두 배가 되어 $4|X|$ 가 된다.

$$T = (T_{request} + T_{response}) * I \approx 2I = 4|X|, T_{request} \text{은 요청을 위한 트랜잭션의 수, } T_{response} \text{는 응답을 위한 트랜잭션의 수}$$
식 10

사용자와 시스템 사이의 과도한 네트워크 트래픽을 줄이기 위하여, 이동 에이전트는 적응적 보안 등급 알고리즘을 사용한다. 즉, 이동 에이전트가 사용자와의 상호 작용을 직접 수행함으로써 단지 이동 에이전트의 상호 작용이 끝난 후의 한번의 비동기 Trap만이 요구된다. 따라서 제안된 모델은 SNMP 기반의 모델(대략 4X가 요구될 것이다.)에 비하여 상호 작용의 수를 상당히 줄일 수 있을 것이다.

4.3 기존 보안관리 시스템과의 비교

자원의 중요도를 고려함으로써 기존의 ISM(Integrated Security Management) 모델에 비하여 불필요한 자원 소모를 줄일 수 있는 보안 관리 모델이 제안된 바 있다[6]. 이 모델은 자원의 중요도에 따라서 보안 서비스를 차등화하여 제공함으로써 효율성을 높일 수 있었다. 그러나 보안을 고려하는데 있어서 자원의 중요도 이외에도 물리적인 위치, 공격 수준, 취약점 수준과 같은 많은 변수들이 존재한다. 본 논문에서 제안된 모델은 다중 변수를 고려할 수 있는 알고리즘을 사용함으로써 보다 안전하고 효율적인 보안 관리를 제공할 수 있을 것이다.

5. 결론

기존의 보안 모델은 정적인 정책 결정 방법을 사용함으로써 네트워크 환경의 동적인 변화에 대응할 수 없으며, 단일 변수에 의존한 보안 모델은 유연한 보안 관리가 어렵다.

본 논문에서 다양한 네트워크 환경의 동적인 변화에 따라서 동적으로 정책들을 적용할 수 있는 보안 관리 모델을 제안하였다. 제안된 모델의 주요 특징들은 다음과 같다.

먼저, 이동 에이전트에 의한 사용자와의 상호 작용 후의 한 번의 비동기 Trap만이 요구된다. 따라서 SNMP 기반의 모델에 비해서 상호 작용의 수를 줄일 수 있다. 그리고 기존 접근 방법에서 단일 변수만 고려하는 것과는 달리 다중 변수를 다룬다. 따라서 제안된 모델은 이기종 네트워크 환경에서 보다 유연한 보안 관리를 수행할 수 있다. 마지막으로, 기존의 보안 관리 모델은 보통 정적인 의사 결정 방식을 사용한다. 반면에, 제안된 모델은 환경의 변화에 따라서 적절한 암호 알고리즘과 프로토콜들을 적응적으로 적용함으로써 자원의 낭비를 줄일 수 있다.

그러나 이동 에이전트와 사용자 사이의 많은 상호 작용은 여전히 문제가 될 수 있다. 향후 연구로서 이러한 상호 작용을 줄이기 위한 방안에 대하여 연구하고자 한다. 또한 다양한 암호화 기법과 프로토콜들의 암호학적 강도와 안전성에 대한 평가를 수행함으로써 보다 정량적인 보안 관리를 수행할 수 있는 방안에 대해서 연구하고자 한다.

참고문헌

- [1] R.L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, New York, NY, 1976.
- [2] D. Winterfeld, von and W. Edwards, *Decision Analysis and Behavioral Research*, Cambridge, England: Cambridge University Press, 1986.
- [3] L. Martignon and U. Hoffrage, *Why Does One-Reason Decision Making Work?*, In *Simple Heuristics That Make Us Smart*, Oxford University Press, New York, 1999, pp. 119-140.
- [4] W. Stallings, *SNMP, SNMPv2, and RMON: Practical Network Management*, 3rd ed., Reading, MA: Addison-Wesley, 1999.
- [5] *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, <http://www.ietf.org/rfc/rfc1213.txt>
- [6] *Structure of Management Information Version 2 (SMIv2)*, <http://www.ietf.org/rfc/rfc2578.txt>
- [7] D.S. Kim and T.M. Chung, "A Design of Preventive Integrated Security Management System Using Security Labels and a Brief Comparison with Existing Models," Int'l Conf. on Computational Science and Its Applications(ICCSA2004). LNCS 3043, Springer Verlag. 2004, pp. 183-190.

- [8] D. Gavalas et al., "*Hierarchical network management: a scalable and dynamic mobile agent-based approach*," Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 38, Issue 6, 2002, pp. 693-711.
- [9] 이정우, 윤완오, 신광식, 최상방, "SNMP와 이동에이전트의 해석적 모델 및 성능 평가", 한국통신학회 논문지, 28권, 8B호, 2003, 716-729쪽.
- [10] *The S/Key One-Time Password System*,
<http://www.ietf.org/rfc/rfc1760.txt>
- [11] *A One-Time Password System*, <http://www.ietf.org/rfc/rfc2289.txt>
- [12] *American National Standard DSTU X9.59 Electronic Commerce for Financial Service Industry: Account Based Secure Payment Object*, 2000.
- [13] Albert Levi. Certin K. Koc, "*CONSEPP: Convenient and Secure electronic payment Protocol based on X9.59*," Proc. of 17th Annual Computer Security Application Conference, 2001, New Orleans, Louisiana.
- [14] Jongwoo Chae, Ghita Kouadri Mostéfaoui, and Mokdong Chung, "*An Adaptive Security Model for Heterogeneous Networks Using MAUT and Simple Heuristics*," Int'l Conf. on Computational Science and Its Applications(ICCSA2004). LNCS 3046, Springer Verlag. pp.958-967, 2004.
- [15] 채종우, 서정철, 정목동, "*Adaptive SSL/TSL Security Service For Ubiquitous Computing Environment*," 한국정보과학회 영남지부 학술

발표논문집, 11권 1호, 2003, pp.133-139.

- [16] *The SSL Protocol Version 3.0*,
<http://wp.netscape.com/eng/ssl3/draft302.txt>.
- [17] *The TLS Protocol Version 1.0*, <http://www.ietf.org/rfc/rfc2246.txt>.
- [18] Jongwoo Chae, Jungchul Seo, Mokdong Chung, "*Estimating User's preference In Ubiquitous Computing Using Multi-Attribute Theory*,"
Proc. of The 2003 International Symposium on Advanced Engineering, Korea, Nov. 2003, pp. 222-227
- [19] 김환조, 채종우, 정복동, "*Design and Implementation of DRM-based Contents distribution System using X9.59*," 멀티미디어학회 학술발표 논문집, 6권, 1호, 2003, pp.771~774.
- [20] Bouncy Castle 1.25 API Specification, <http://www.bouncycastle.org/docs/docs1.4/index.html>.