개선된 CGA(Modified CGA)를 이용한 계층적 애드 혹 네트워크에서의 주소 자동 설정 및 전자 서명 제공 방안

(Design of Modified CGA for Address Autoconfiguration and Digital Signature in Hierarchical Ad Hoc Network)

이 혜 원 † 김 국 보 †† 문 영 성 †††
(Hyewon K. Lee) (Gukboh Kim) (Youngsong Mun)

요 약 IPv6 워킹 그룹에서 표준화된 CGA(Cryptographically Generated Addresses)는 링크상에서의 주소 변조 및 주소 도난 문제를 해결하고 전자서명을 제공하기 위해 제안되었으나 키 충돌이라는 문제가 발생할 수 있어서 이를 해결하기 위해 SEC(SECurity parameter) 필드를 도입하여 높은 보안이 필요한 경우에는 높은 SEC 값을 적용함으로써 키 충돌 확률을 감소시킨다. 하지만 SEC 값이 증가함에 따라 CGA 생성 시간이 무제한으로 증가하기 때문에 무선 환경에서 SEC 값이 높은 CGA를 적용하는 것은 불가능하다. 또, 낮은 SEC 값을 적용하는 경우 키 충돌은 높은 확률로 발생한다. 따라서, 본 논문에서는 계층적 애드 혹 환경에 적합한 개선된 CGA(MCGA: Modified CGA)를 제안한다. 제안되는 MCGA는 CGA에 비해 생성 시간이 매우 짧고 CGA와 마찬가지로 매우 작은 오버헤드로 전자 서명을 제공하며 계층적 네트워크 환경에서 사용함으로써 키 충돌 문제를 해결한다. MCGA는 계층적 애드 혹 환경뿐만 아니라 일반 IPv6 네트워크에서도 적용이 가능하다.

본 논문에서는 먼저 수학적 모델을 통해 MCGA와 CGA의 생성 시간을 분석하고 시뮬레이션을 통해 CGA와 MCGA의 생성시간을 측정하여 MCGA가 SEC 값이 0인 경우의 CGA에 비해 생성 시간이 평균 3.3배, 그리고 SEC 값이 1인 경우에는 평균 68,000배 짧다는 것을 보인다. 특히 SEC 값이 3 이상인 경우 애드 혹 환경뿐만 아니라 일반 네트워크에서도 부적절하다는 것을 증명한다.

키워드 : 개선된 CGA, 계층적 애드 혹 네트워크, 전자서명, 주소 변조, 주소 도난, SHA, MD5, CGA (Cryptographically Generated Addresses), SEC(SECurity parameter)

Abstract The CGA proposed by IETF working group prevents address spoofing and stealing and provides digital signature to users, but key collision problem arises. To solve this critical problem, the CGA defines the SEC field within address format, which is set to high value when high security is required and vice versa, but the CGA faces a dilemma between security and the processing time. As SEC value increases, the processing time to generate the CGA grows dramatically while key collision ratio increases if low SEC value is applied to the CGA. We propose modified CGA (MCGA) that has shorter processing time than the CGA and offers digital signature with small overheads. To solve key collision problem, we employ hierarchical ad hoc network. The MCGA is applicable to IPv6 networks as well public networks.

In this paper, we design a mathematical model to analyze the processing time for MCGA and CGA first and evaluate the processing time via simulations, where the processing time for MCGA is reduced down 3.3 times when SEC value is set to 0 and 68,000 times when SEC value is set to 1. Further, we have proved that the CGA is inappropriate for both ad hoc networks and IPv6 networks when the SEC field is set to more than 3.

Key words : modified CGA(MCGA), hierarchical ad hoc network, digital signature, address spoofing, address stealing, SHA, MD5, CGA(cryptographically generated address), SEC(SECurity parameter)

† 학생회원 : 숭실대학교 컴퓨터학과
kerenlee@cherry.ssu.ac.kr
†† 정 회 원 : 대진대학교 컴퓨터공학과 교수
kgb@daejin.ac.kr
††† 종신회원 : 숭실대학교 컴퓨터학과 교수
mun@computing.ssu.ac.kr
논문접수 : 2005년 8월 2일
심사완료 : 2005년 11월 7일

1. 서 론

이동 애드 혹 네트워크는 기지국에 의존하지 않는 다중 홉 무선 네트워크이다. 이 기술은 현재 동적인 주소 할당을 위해 주로 사용되는 DHCP 프로토콜[1]이나 경로 설정을 위해서 라우터를 사용하지 않고, 통신에 참가하는 노드가 라우터 역할을 함으로써 네트워크를 구성한다. 이동성을 지원하는 무선 네트워크 또는 유선 네트워크와 이동 애드 혹 네트워크의 가장 큰 차이는 하부 인프라에 의존하지 않으면서 네트워크 토폴로지가 빠른 속도로 연속적으로 바뀐다는 점이다.

현재 DSR[2], AODV[3], TBRPF[4] 등의 라우팅 프로토콜이 제안되었지만 이들은 모두 근원지에서 목적지까지의 최적화 또는 최단 경로를 찾기 위한 라우팅 프로토콜만을 기술하고 있어서 네트워크 형성 이전에 노드의 설정이 되어 있다고 가정하고 있다. 이를 보완하기 위해서 MANETConf[5], 자동 노드 설정 프로토콜[6] 및 예언 주소 할당 알고리즘[7] 등이 제안되고 있다. 특히 [5]에서는 단일 계층 구조에서의 노드 주소 할당 및 중복성 회피를 위한 방안을 제시하고 있는 반면 [6]에서는 계층적 구조에서의 노드 설정 및 주소 할당에 대하여 제안하고 있으나 이들은 어떤 형식의 주소를 생성하여 사용할 것인지에 대해서 고려하지 않는다.

IETF의 워킹그룹 SEND에서 제안하고 있는 CGA는 작은 오버헤드로 전자서명을 제공하고 있어서 메모리나 처리 능력이 일반 네트워크 환경보다 열등한 애드 혹 환경에서 사용할 경우 키 공개 및 교환을 위한 메시지 교환을 최소화 한다. 하지만 CGA의 보안성을 증가시킬 경우 생성 시간이 증가하여 일반 노드가 주소를 자동 생성하여 통신에 참가하기까지 오랜 시간동안 휴지상태에 있어야 한다는 문제가 있다. 따라서 본 논문에서는 [8]에서 제안하는 방안을 수정 및 보완함으로써 애드 혹 네트워크에서 적절히 사용할 수 있는 주소 형식인 개선된 CGA(MCGA)를 제안한다.

MCGA는 CGA에 비해 생성 지연이 매우 짧고 CGA의 가장 큰 특징인 전자서명을 제공한다. CGA 생성에서 가장 큰 문제점인 키 충돌 문제를 해결하기 위해서 본 논문에서는 균일 계층의 네트워크 구조가 아닌 계층적 네트워크 구조를 기반으로 하고 있으며 이를 위해 [6]에서 제안하는 에이전트와 컨시그너로 구성된 2 계층 구조를 사용한다. 이 구조는 또한 빠르고 용이한 주소 중복성 검사를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 애드 혹 네트워크에서의 주소 할당 프로토콜과 새로운 주소 형식인 CGA에 대해 살펴보고, 3장에서는 주소 자동 설정을 위해서 본 논문에서 제안하는 주소 형식 및 세부 고려 사항을 기술한다. 4장에서는 성능평가를 위한 환경을 설명하고 CGA와 MCGA간의 생성 지연을 평가한 후 이를 분석하고, 5장에서 향후 과제를 남기면서 결론을 짓는다.

2. 관련 연구

이동 애드 혹 네트워크에서의 주소할당 문제를 해결하기 위해서 현재까지 충돌감지(conflict-detection) 주소할당 알고리즘[9], 충돌회피(conflict-free) 주소할당 알고리즘[10], 최선(best-effort) 주소할당 알고리즘[11], 예언(prophet) 주소할당 알고리즘[7] 등이 제안되었다. 특히 [5]에서 제안하는 방식은 최선 주소 할당 알고리즘의 대표적인 방식으로 주소 할당시 중복성 검사를 위한 메시지 교환의 오버헤드가 크다. [6]은 [5]의 주소 할당시의 높은 오버헤드 문제를 해결하기 위해서 계층적 네트워크 구조를 기반으로 하여 빠른 주소할당 및 용이한 중복성 검사를 제공한다. 이들은 모두 낮은 오버헤드를 갖는 주소 할당 및 충돌 문제 회피 혹은 해결을 위한 알고리즘을 제안하고 있으나 어떤 방식으로 주소를 생성할 것인지에 대해서 고려하지 않는다.

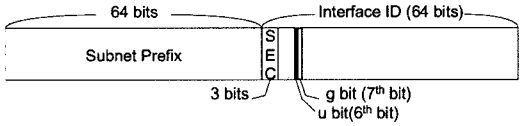


그림 1 CGA 형식

CGA[8]는 통신에 참가하려는 IPv6 노드가 랜덤 변수와 자신이 생성한 공개키, 서브넷 프리픽스 및 여러 변수를 조합하여 SHA 해시 함수[11]에 입력하여 결과 160 비트 중 선두 64 비트만을 선택하여 특정 조건을 만족하는지 여부에 따라 자동으로 주소를 생성할 수 있는 형식이다. 특히, 주소 생성시 사용된 공개키를 사용하여 작은 오버헤드로 전자서명이 가능하다. 그림 1에서 CGA 형식을 볼 수 있다. CGA는 64 비트의 키 값만을 쓰기 때문에 그림 4에서 보이는 바와 같이 충돌 문제가 있어서 원래의 사용자가 사용한 키 쌍이 아닌 다른 키 쌍을 사용해서 동일한 CGA를 생성하는 확률 즉, 동일한 CGA를 생성 가능하게 만드는 다른 키 쌍의 발견 확률이 높다.

이를 해결하기 위해 3 비트 길이의 SEC 필드[1] 값을 조정하여 유효한 주소 및 유효하지 않은 주소로 분류, 유효한 주소의 경우에만 실제 노드의 인식자로

1) SEC 필드는 CGA의 64 비트 인터페이스 아이디 중 첫 번째 3 비트에 정의된 것으로 0~7사이의 값을 갖는다

사용할 것을 제안하고 있으나 SEC 값이 증가하는 경우 주소 키 충돌율은 감소하지만 생성 시간이 무한대로 증가하기 때문에 무선 환경에서 SEC 값이 높은 CGA를 적용하는 것은 불가능하다. 이와 반대로 낮은 SEC 값을 적용하는 경우 키 충돌은 높은 확률로 발생한다.

CGA 생성 절차는 다음과 같다. SEC×16 비트와 MODIFIER[2]의 AND 연산의 결과치가 일련의 0이 나올 때까지 MODIFIER를 순차적으로 1씩 증가시켜 유효 MODIFIER를 구한다. 유효 MODIFIER와 다른 변수들을 조합하여 이를 SHA 함수에 입력하여 얻은 결과 중 왼쪽 64 비트를 취하여 그림 1과 같은 형식의 주소를 생성한다. SEC 값이 증가할수록 비교해야 하는 비트 수가 증가하기 때문에 유효 MODIFIER를 찾는 시간이 증가한다.

3. 개선된 CGA(Modified CGA) 제안

CGA는 충돌 문제를 회피하기 위해 SEC 값을 조정하여 유효한 주소 및 유효하지 않은 주소로 분류, 유효한 주소의 경우에만 실제 노드의 인식자로 사용하였다. 이에 반해 본 논문에서 제안하는 MCGA는 계층적 네트워크 환경에서 적용함으로써 충돌 문제를 해결하고 SEC 필드의 유효성 검사를 제거함으로써 주소 생성시까지의 지연을 단축시킬 수 있어서 애드 혹 환경에 적합하다. 이 외에도 CGA의 기본적인 특성을 그대로 상속받기 때문에 부가적인 메시지 교환이 없이 전자서명을 제공한다.

본 논문에서는 계층적인 네트워크 구조를 위해서 [6]에서 제안하는 에이전트와 컨시그너로 구성된 2 계층 구조를 사용한다. 에이전트는 단위 네트워크(MUNIT)를 관리하는 노드이며 하나의 MUNIT에는 하나의 에이전트와 하나 이상(에이전트 포함)의 컨시그너로 구성된다. 컨시그너는 동일한 단위 네트워크(MUNIT)에 대한 주소 정보와 이웃한 에이전트에 대한 정보를 가지고 있으며 각 에이전트는 네트워크 내의 모든 노드에 대한 주소 정보를 가지고 있다.

3.1 MCGA 구조

본 장에서 제안하는 MCGA는 64 비트 길이의 서브넷 프리픽스와 64 비트 길이의 인터페이스 인식자로 구성된다. 서브넷 프리픽스는 노드가 현재 네트워크에서 할당 받은 인식자에 해당되며 인터페이스 인식자는 개별 노드가 생성하는 것으로 일반적으로 랜덤값이나 EUI-64 형식의 주소를 사용하여 생성할 수 있다. 애드 혹 환경에서의 노드는 미리 고정된 인터페이스 카드를 가지고 있다고 가정할 수 없기 때문에 모든 노드가 EUI-64 형식을 사용하여 주소를 생성한다는 것은 불가능하다. 따라서 본 논문에서는 인터페이스 카드의 인식자를 사용할 수 있는 경우에는 NIC(Network Interface Card) 번호를, 그 이외의 경우에는 랜덤 값을 사용한다. 서브넷 프리픽스를 위해서는 로컬 주소에 해당하는 FE80::/64를 가정한다.

3.2 MCGA 생성

CGA가 인터페이스의 생성을 위해 SHA 함수를 사용한 반면 MCGA는 MD5[12]함수를 사용하는데 이는 MD5 함수의 알고리즘이 더 간단해서 수행 시간이 더 짧기 때문이다.[3] 이외에 키 생성을 위해서는 RSA 알고리즘을 사용한다. MCGA 생성 알고리즘을 그림 2에서 보이고 있으며 그 절차는 아래와 같다.

(1) RSA 알고리즘을 사용하여 키 쌍을 생성한다.
(2) 난수를 발생하거나 NIC 주소를 사용하여 MODIFIER를 구하고 중복회수를 0으로 설정한다.
(3) MODIFIER와 중복 회수 및 사용자의 공개키 등의 변수를 조합하여 MD5 함수의 입력 값으로 사용한다.
(4) MD5 함수의 128 비트 출력 값 중 왼쪽 64 비트를 취하여 이를 인터페이스 아이디로 설정한다.
(5) 인터페이스 아이디의 여섯 번째 및 일곱 번째 비트를 0으로 세팅한다.
(6) 서브넷 프리픽스와 인터페이스 아이디와 조합한다.
(7) 중복성 검사 후 중복이 아닌 경우 (6)에서 구한 주소를 인터페이스에 할당한다. 중복인 경우 중복 회수를 1 증가시켜 (3) 단계부터 다시 시작한다. 중복 회수가 3이 되는 경우 (2) 단계부터 시작한다.

3.3 주소 중복 문제에 대한 고려사항

네트워크에 진입한 노드는 3.2절에서 설명한 바와 같이 MCGA를 생성하고 중복이 없는 경우 자신의 인터페이스에 할당할 수 있다. 이때 노드는 가장 가까이 위치한 에이전트에게 중복성 검사를 요청하는데 요청을 받은 에이전트는 자신의 주소 테이블 검색 후 중복이 아닌 경우 우선적으로 중복이 아님을 알리고 다른 에이전트에게 중복 여부를 묻는다[6]. 중복이 없는 경우 에이전트는 요청한 노드에게 최종 긍정 응답 메시지를 전송한다. 이와 같이 주소 중복성 검사는 중복 검사 요청을 개시한 후 에이전트로부터 두 개의 긍정 응답 메시지를 받아야 종료된다. Opti-DAD(Optimistic Duplication Address Detection)[13]를 사용하는 경우 첫 번째 응답

2) MODIFIER는 인터페이스 아이디 생성을 위해 발생된 난수이다.

3) 일반적으로 160 비트 키 값을 갖는 SHA 함수가 128 비트 키 값을 갖는 MD5 함수보다 더 강력한 보안을 제공한다고 알려져 있으나 CGA는 왼쪽 64 비트만을 취하기 때문에 SEC 필드의 값을 높게 잡더라도 충돌률이 높다.

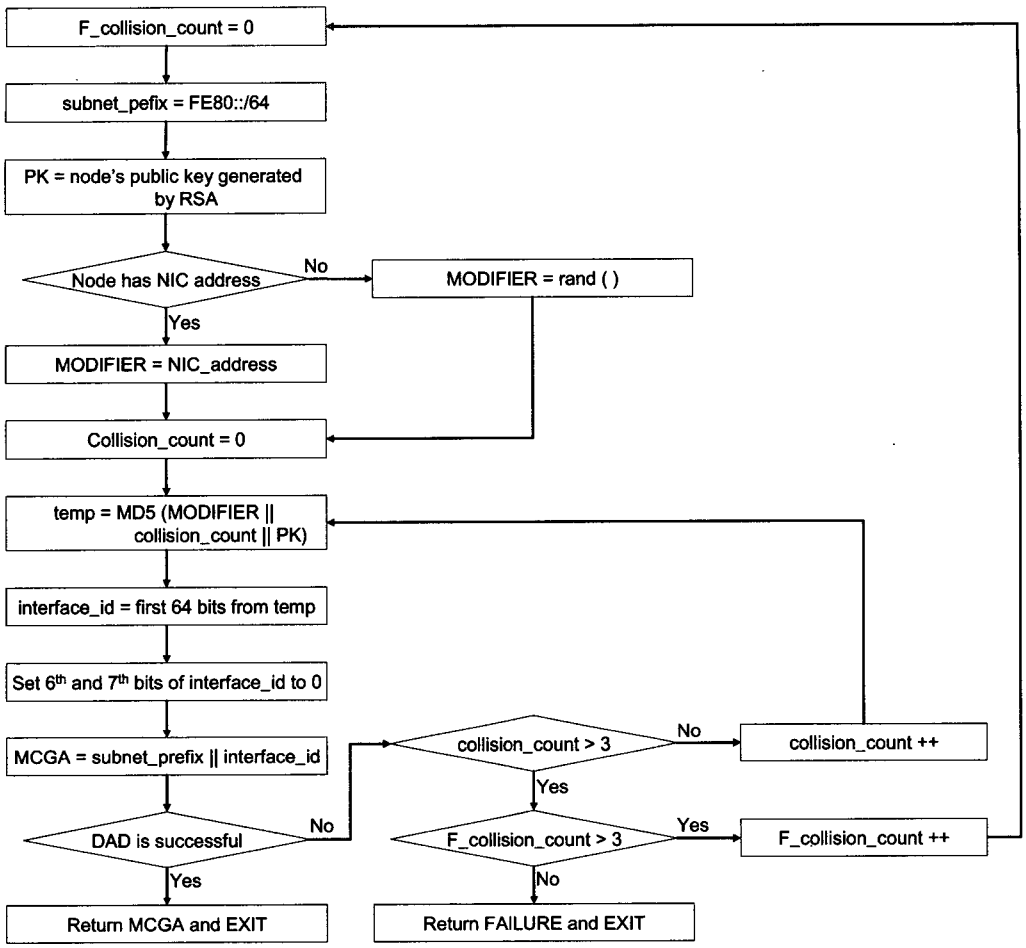
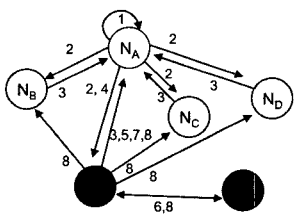


그림 2 MCGA 생성 알고리즘

메시지를 받은 후 노드는 통신을 개시할 수 있다. 이상의 절차를 그림 3에서 보이고 있다.

네트워크에 n 개의 노드가 있을 때 n 개의 MCGA 생성시 중복이 없을 확률은 ${}_{2^{64}}P_n/(2^{64})^n$ 과 같아서 중복



1. New node (N_A) builds MCGA
2. N_A broadcasts neighbor solicitation message
3. Neighbored nodes reply to N_A
4. N_A selects proper responder (agent, AG_1) and requests address duplication check
5. AG_1 checks its address table and sends acknowledgement to N_A
6. AG_1 requests other agents to check whether on-pending address is duplicated
7. AG_1 sends acknowledgement message to N_A
8. AG_1 broadcasts update message to MUNIT immediately and to agents periodically.

그림 3 노드의 네트워크 진입 후 중복성 검사 수행 절차

이 한 번 이상 나올 확률은 수식 (1)과 같이 나타낼 수 있다. 예를 들어 1000 개의 주소를 생성한다고 했을 때 주소 중복률은 0이다. 특히, 계층적 구조의 네트워크에서는 논리적으로 상위 계층에 있는 노드가 네트워크 주소 자원에 대한 상태 정보를 가지고 있어서 1홉 거리에 있는 하위 계층의 노드와 상위 계층 노드간의 중복성 검사만으로 중복 여부를 따질 수 있다. 따라서 본 논문에서는 opti-DAD[13]를 적용한다.[4] 즉, DAD가 완료되기 이전에 노드는 임시 주소를 사용하여 통신에 참여할 수 있으며 통신 개시까지의 지연을 줄일 수 있다. 주소 생성의 중복률이 매우 낮다고 하더라도 할당되지 않은 상태의 주소를 상이한 노드가 동시에 선택하여 주소 중복 검사를 하는 중복이 발생할 수 있는데 중복 검사를

4) Opti-DA의 경 주소 중복 검사가 완전히 끝나기 전에 검사 중인 주소를 사용하기 때문에 중복이 발생하는 경우 주소를 다시 생성하여 중복성 검사를 재개해야 한다. 따라서 [13]은 중복율이 낮은 네트워크에서 Opti-DAD를 적용할 것을 권고한다.

먼저 개시한 에이전트가 검사중인 주소에 대한 우선순위를 갖게 할 수 있다.

$$1-\frac{2^{64}P_n}{(2^{64})^n} \quad (1)$$

3.4 MCGA 사용을 위한 각 노드의 동작

MCGA 주소를 사용하여 통신을 하는 경우, 모든 메시지는 MCGA를 생성한 모든 변수(MODIFIER, 공개키, 중복 계수)를 포함하고 있어서 수신측 노드가 수신된 메시지에 대한 주소 유효 여부를 따지게 되며 유효한 경우에 한해 메시지를 상위계층에 전달하거나 다음 노드에게 전달한다. 즉, 수신자는 메시지에 포함된 여러 파라미터를 사용하여 해당 MCGA가 정당하게 생성되었는지 여부를 확인할 수 있다. 이때 전달자의 역할을 수신자가 일반 노드인 경우 주소 검사 없이 바로 전달할 수 있다.

각 노드는 통신을 하는 상대(목적지) 노드에 대해 {address state, address, physical address, registered time, public key, lifetime, number of hop count} 정보를 저장한다. 각 필드에 대한 세부적인 설명을 표 1에서 보이고 있다. 이 정보는 반드시 유효한 라이프 타임 필드를 가지고 있어야 하며 유효하지 않은 레코드는 반드시 삭제해야 한다. 또, 각 노드가 저장하고 있는 정보와 일치하지 않는 정보로 메시지가 생성되어 수신된 경우 이를 드랍한다.

표 1 각 애드 혹 노드가 상대 노드에 대해 저장하고 있는 정보

필 드	설 명
Address state	할당, 비할당, 휴지 상태를 나타냄
Address	상대 노드의 IP 주소
Physical address	상대 노드의 물리 주소
Registered time	할당, 비할당, 휴지 상태에 진입한 시간
Public key	상대 노드의 공개키
Lifetime	해당 레코드의 유효 시간
Number of hop count	홉 수

3.5 MCGA를 사용한 전자서명

MCGA는 사용자의 공개키를 포함하고 있어서 기본적으로 전자 서명을 제공한다. 즉, 자신의 키로 전송하려는 메시지를 암호화하여 이를 전자서명으로서 사용할 수 있으며 수신자는 MCGA를 할당 받은 정당한 사용자가 전송했음을 확인할 수 있다. 메시지에 담긴 내용이 중요할 때 송신 노드는 수신 노드의 공개키를 사용해서 암호화 후 전송할 수 있다.

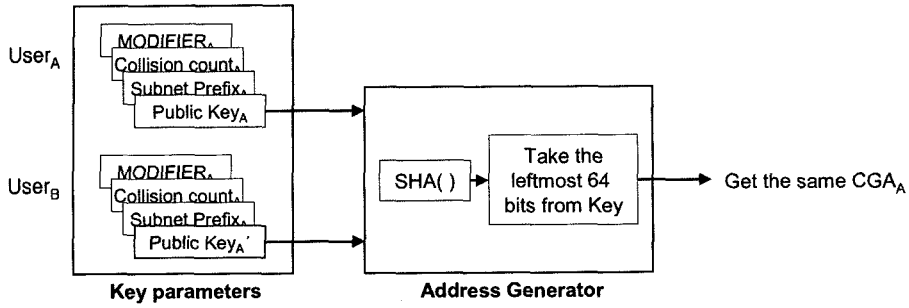
3.6 키 충돌 보안 측면에서의 고려사항

일반적으로 해시 함수의 키 충돌률은 0%이고 SHA 해시 함수는 다른 해시 함수에 비해 안전하다고 알려져 있으나 CGA는 해시 함수의 충돌이라는 큰 문제점을 안고 있다. 이때 산술적으로 64 비트 인식자 하나당 2^{96} 개의 주소가 사상 된다. 예를 들어 해시 함수(h)와 서로 다른 두 개의 입력 값 m_1, m_2가 있다고 할 때 $h(m_1)\neq h(m_2)$가 성립하지만 SHA 함수의 출력 값 중 선두 64 비트만을 선택했을 때는 충돌이 발생할 수 있으며 이때의 충돌률은 해시 함수의 '생일문제(birthday problem)'로 증명할 수 있는데 대략 0.63이 된다[14].

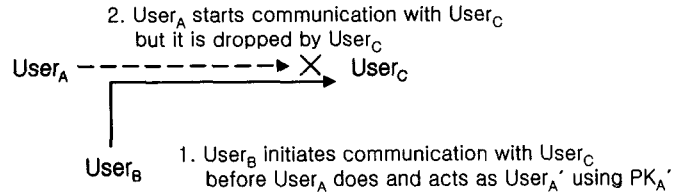
이와 같이 다른 메시지를 입력 값으로 하여 동일한 64 비트 값을 생성할 확률이 매우 높으므로 원 키에 대응되는 거짓 키를 생성할 확률이 매우 높다고 말할 수 있으며 거짓 키를 생성하여 사용하는 경우 주소 변조 및 주소 도난 공격이 가능하다. 예를 들어 그림 4에서 보이는 바와 같이 임의의 노드(N_A)가 자신의 공개키 PK_A로 생성하여 사용하고 있는 주소(CGA_A)를 다른 노드(N_B)가 PK_A'로 CGA_A를 생성하여 사용할 수 있다. 이렇게 생성된 주소(CGA_A) 및 파라미터를 사용하여 N_B가 노드 N_C와 메시지를 교환한다고 했을 때 N_C는 N_B로부터 받은 메시지에 대하여 주소 적절성 검사를 수행 후 오류가 없기 때문에 N_B가 CGA_A를 정당하게 할당받아 사용하는 노드로 인식할 것이다. CGA_A의 원 사용자(N_A)의 개인키가 노출되지 않은 이상 N_A가 생성하는 전자서명을 흉내내거나 N_A의 공개키로 암호화한 데이타를 해독할 수는 없으나 주소 도난을 통해 상대 노드를 잘못된 통신으로 유도할 수 있다.

이와 같은 문제를 피하기 위해 [8]에서는 SEC 필드를 사용하여 SHA 해시 함수 키 값 중 최대 112 비트를 사용하여 유효성 검증을 제안하고 있으나 높은 SEC 값을 선택시 주소를 생성하는 노드 측의 적절한 MODIFIER를 찾을 때까지의 지연이 길어진다. 높은 SEC 값을 사용한다고 하더라도 단일 계층 네트워크에서 CGA를 원 사용자의 키에 대응되는 키를 찾게 되면 동일한 문제가 발생한다. 이는 CGA를 생성하는 노드들의 상태 정보를 CA 혹은 논리적으로 상위 계층에 있는 노드가 가지고 있지 않기 때문이다. 이 외에 CGA 기법은 바인딩 정보나 라우팅 정보 등을 의도적으로 전달함으로써 발생하는 통신 방해 문제를 회피하기 위해 각 노드는 키 및 파라미터 정보를 가지고 있지 않아서 메시지 전송시마다 CGA 생성에 따른 파라미터를 교환하기 때문에 메시지의 길이가 길어지고 메시지마다 주소가 적절한지 여부를 계산해야 한다는 문제가 발생한다.

이에 반해 계층적 애드 혹 네트워크 모델에서 MCGA 주소를 사용하는 경우에는 주소에 대한 상태 정보를 상위 노드 혹은 같은 영역 내에 있는 노드가 가지고 있기 때문에 이와 같은 공격을 막을 수 있다. 예를 들어 임의



(a) 입력값 m_1, m_2 ($m_1 \neq m_2$)에 대해 동일한 CGA를 생성



(b) 키 충돌로 인한 공격 발생

그림 4 키 충돌로 인한 공격

의 노드(N_A)가 자신의 공개키 PK_A로 생성하여 사용하고 있는 주소($MCGA_A$)를 다른 노드(N_B)가 PK_A'로 $MCGA_A$를 생성하여 메시지를 전송한다고 하자. N_A와 N_C가 같은 영역(MUNIT) 내에 있는 노드인 경우 계층적 모델에서는 모든 노드가 동일한 영역 내의 주소 자원 정보를 모두 알고 있기 때문에 N_B가 $MCGA_A$를 사용하여 N_C에게 메시지를 전송한다면 주소 중복성 검사시 오류가 발생하며 N_B를 공격자로 인식한다. 또, N_A와 N_C가 다른 영역에 속한다고 했을 때 전달자 역할을 하는 에이전트가 N_B로부터의 메시지 수신 후 주소와 공개키 간의 바인딩이 부적절하다는 것을 알 수 있어서 N_B로부터의 공격을 봉쇄할 수 있다. 이는 임의의 주소에 대한 선점권 상태 정보를 이웃 노드 혹은 에이전트가 가지고 있기 때문이다.

계층적인 네트워크 구조에서 상위에 있는 노드가 하위 노드에 대한 주소 및 상태정보를 가짐으로써 주소 중복성 검사시 허위로 응답하여 주소 할당을 방해하거나 메시지를 임의로 드랍하는 문제가 발생할 수 있다. 먼저, 그림 2의 알고리즘 수행이 실패로 끝나는 경우 주위의 다른 에이전트[5]를 선택하여 주소 중복성 검사를 요청함으로써 주소 할당 방해 문제를 해결할 수 있다. 또, 에이전트가 메시지를 임의로 드랍하는 경우는 이웃 노드가 이를 감지할 수 있어서 오동작하는 에이전트로 인식하여 새로운 노드를 에이전트로 선택함으로써 해결할 수 있다. 이 외에 네트워크 내의 임의의 노드가 시도할 수 있는 replay 공격은 타임스탬프를 사용함으로써 막을 수 있다[8].

4. 성능 평가

성능 평가를 위해서 본 논문에서는 수학적인 모델을 먼저 설계한 뒤 MCGA가 CGA에 비해 애드 혹 환경에서 적절함을 보이고 시뮬레이션을 통해 MCGA와 CGA의 생성 시간을 비교 분석한다.

4.1 모델링

[8]에서 제안하고 있는 CGA 생성을 위해 필요한 처리 시간은 유효한 MODIFIER를 구하기 위한 시간과 인터페이스 아이디의 생성 및 중복성 검사를 위한 시간의 합으로 나타낼 수 있다. 주소 생성시 중복이 m 회 있다고 할 때 CGA 생성을 위한 지연(L_{CGA})은 수식 (2)와 같다. 이에 반해 MCGA 생성을 위해서는 CGA와 같이 유효성을 검증하기 위한 지연이 없어 단순히 난수를 생성하여 MODIFIER를 선택하는 시간과 인터페이스 아이디의 생성 및 중복성 검사를 위한 시간의 합으로 나타낼 수 있다. 주소 생성시 중복이 m 회 있다고 할 때 MCGA 생성을 위한 지연 (L_{MCGA})은 (3)과 같다. 각 수식의 변수에 대한 설명은 표 2에서 보이고 있다.

5) 네트워크에 노드가 진입했을 때 그림 2에서 보이고 있는 것처럼 주위 노드에게 이웃 검색 메시지를 브로드캐스트하고 여러 에이전트로부터 응답 메시지를 받는 경우 적절한 에이전트를 선택하여 주소 중복성 검사를 개시하기 때문에 주소 생성 중에 있는 노드는 이웃 에이전트에 대한 정보를 가지고 있다.

$$L_{CGA} = \left\lfloor \frac{m}{2} + 1 \right\rfloor l_{MOD} + (m+1)(l_{SHA} + l_{DAD}) \quad (2)$$

$$L_{MCGA} = \left\lfloor \frac{m}{2} + 1 \right\rfloor l_{RV} + (m+1)(l_{MD5} + l_{DAD}) \quad (3)$$

표 2 시스템 파라미터

변수	설명
l_{DAD}	중복 검사를 위한 지연
l_{RV}	난수를 생성하기 위한 처리 시간
l_{MOD}	적절한 MODIFIER를 구하기 위한 처리 시간
l_{SHA}	SHA 함수 수행 시간
l_{MD5}	MD5 함수 수행 시간
m	주소 생성시 중복 회수

주소 생성 후 중복성 검사를 위한 지연, l_{DAD}는 네트워크에서 정의하고 있는 네트워크 크기에 의해 결정된다. 이를 구하기 위해서 [15]의 한 노드에서 목적지 노드에 데이타 송신시의 수렴 시간을 변형하여 수식 (4)와 같이 나타낼 수 있다. 데이타 교환시 가장 긴 경로는 네트워크의 각 최단거리 경로의 최대값 혹은 네트워크 지름(l_d)이 된다. 바이트 길이 데이타를 초당 b비트를 전송하는 대역폭을 갖는 단일 링크상의 전송 시간은 $8s/b$이다. 이 외에 네트워크에서 데이타를 전송하기 이전에 있는 처리 지연을 d ms 혹은 $d+r\,(0 \le r \le 1)$ ms 라 가정할 때 l_{DAD}는 (4)와 같이 나타낼 수 있다.[6]

$$2l_d(d + \frac{8s}{b}) \le l_{DAD} \le 2l_d(d + r + \frac{8s}{b}) \quad (4)$$

4.2 시뮬레이션

시뮬레이션을 위해서 사용된 시스템의 CPU는 Pentium 4 3GHz이고 메모리는 1 GB이다. 운영체제는 Linux Kernel 2.4가 사용되었다. 300개의 CGA와 MCGA를 순차적으로 생성한다고 했을 때의 수행 시간 변화를 그림 5에서 보이고 있다. SEC 값은 0, 1, 2로 고정되어 측정되었다.

300개의 MCGA 생성 평균 수행 시간은 4.77μs이다. 이에 반해 SEC 값이 0인 300개의 CGA 생성 평균 수행시간은 15.57μs이다. 평균 수행 시간의 측정 단위는 μs로 무척 작은 단위이나 본 시뮬레이션에 얻은 주소 생성 지연은 애드 혹 노드보다 성능이 매우 좋은 시스템에서 수행한 것으로 이보다 처리 속도 및 메모리 용량이 작은 애드 혹 노드에서의 처리속도는 느려질 것으로 예상할 수 있다. 예를 들어 400 MHz의 애드 혹 노드에서의 수행시간은 실험 결과보다 10배 정도 느려진다. CGA 및 MCGA 모두 300개의 주소를 생성했을 때 주소 중복은 0회 발생했다. 이를 반복 수행하여도 동일한 결과를 산출한다. 본 시뮬레이션에서는 전파지연을 고려하지 않았다.

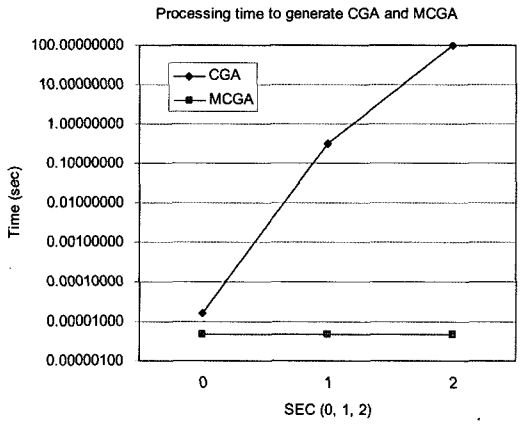


그림 5 CGA 및 MCGA 생성시 시간

SEC 값이 커질수록 주소 생성 지연이 대폭 증가함을 알 수 있다. 이는 수식 (2)에서 MCGA를 생성할 경우 MODIFIER 유효성 검사가 없어서 l_{RV}가 고정 값을 갖는 반면 CGA를 생성할 경우 SEC 값이 증가할수록 마스크 연산을 통해 적절한 MODIFIER를 생성해 내는 시간 l_{MOD} 값이 증가하기 때문이다($l_{MOD} \gg l_{RV}$). SEC 값이 3인 경우 생성시간은 60시간 이상으로 SEC 값이 3 이상인 CGA는 애드 혹 환경에서는 부적절 할 것으로 보인다.

5. 결 론

CGA는 작은 오버헤드로 전자서명을 제공하고 IPv6의 자동 설정으로 인한 주소 변조 및 주소 도난문제를 해결하기 위해 제안되었으나 계층적인 네트워크 구조에서 사용되지 않는 경우 주소 변조 및 도난 문제가 발생할 수 있어서 SEC 필드를 도입하여 높은 보안이 필요한 경우에는 높은 SEC 값을 적용함으로써 키 충돌 확률을 감소시킨다. 그러나 SEC 값의 증가에 따라 보안성이 증가하지만 이와 동시에 CGA 생성 시간이 무제한으로 증가하기 때문에 무선 환경에서 SEC 값이 높은 CGA를 적용하는 것은 불가능하다. 또한 낮은 SEC 값을 적용하는 경우 키 충돌은 높은 확률로 발생한다.

이에 따라 본 논문에서는 계층적 애드 혹 환경에서 적절하게 사용될 수 있는 주소 형식인 개선된 CGA를

6) opti-DAD를 적용했을 때 l_d는 1홉 거리가 된다.

제안하였다. MCGA는 CGA에 비해 짧은 시간 내에 생성되며 계층적 구조에서 사용되어 논리적으로 상위에 있는 노드가 상태정보를 가지고 있어서 주소 변조 및 도난 문제를 해결하기 때문에 일반 네트워크와 비교했을 때 에너지와 보안에 있어 취약한 애드 혹 환경에 적절히 사용될 수 있다.

본 논문에서는 수학적 모델을 통해 MCGA와 CGA의 생성 시간을 분석 후 시뮬레이션을 통해 CGA와 MCGA의 생성시간을 측정하여 SEC 값의 변화에 따른 주소 생성 시간을 보였으며 SEC 값이 3 이상인 경우 애드 혹 환경뿐만 아니라 일반 네트워크에서도 부적절하다는 것을 증명하였다.

참 고 문 헌

[1] R. Droms, "Dynamic Host Configuration Protocol, RFC 2131," IETF, 1997.

[2] D. Johnson, D. Maltz and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," work in progress, IETF, 2003.

[3] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, IETF, 2003.

[4] F. Ogier, F. Templin and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," work in progress, IETF, 2003.

[5] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile ad Hoc Network," Twenty-First Annual Join Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE, 2002.

[6] H. K. Lee and Y. Mun, "Node configuration Protocol based on Hierarchical Network Architecture for Mobile Ad-Hoc networks," ICOIN 2004, Lecture Notes in Computer Science 3090, 2004.

[7] H. Zhou, L. Ni and M. Mutka, "Prophet Address Allocation for Large Scale MANET," Twenty-Second Annual Join Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE, 2003.

[8] T. Aura, "Cryptographically Generated Address," RFC 3972, IETF, 2005.

[9] N. Vaidya, "Duplicate Address Detection in Mobile Ad Hoc Networks," MobiHoc'02, June 2002.

[10] A. Misra, S. Das, A. McAuley and S. Das, "Auto-configuration, Registration, and Mobility Management for Pervasive Computing," IEEE Personal Communication, August, 2001.

[11] D. Eastlake and P. Jones, "US Secure Hash Algorithm," RFC 3174, IETF, 2001.

[12] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, IETF, 1992.

[13] N. Moore, "Optimistic Duplicate Address Duplication for IPv6," work in progress, IETF, 2004.

[14] http://physics.harvard.edu/probweek/sol46.pdf, "the birth problem," Solution Week 46.

[15] J. Kulik, W. Heinzelman and H. Balakrishnann, "Negotiation-Based Protocols for Disseminationg Information in Wireless Sensor Networks," 2002.

이 혜 원

정보과학회논문지 : 정보통신
제 33 권 제 1 호 참조

김 국 보

1984년 서울산업대학교 전자계산학과 학사. 1986년 연세대학교 공학대학원 전자계산학 석사. 1997년 대구 가톨릭 대학교 전자계산학과 박사. 1988년~1990년 해군 중앙전산소장. 1990년~1993년 부경대학교 교수. 1993년~현재 대진대학교 컴퓨터공학과 교수. 관심분야는 소프트웨어 공학, 시스템 분석 및 설계, e-Biz 시스템

문 영 성

정보과학회논문지 : 정보통신
제 33 권 제 1 호 참조