이동에이전트의 자유로운 순회를 보장하기 위한 안전한 베이스캠프에 관한 연구

박종열*, 김홍국*, 오형근**, 박중길**, 이진석**
*광주과학기술원 정보통신공학과
**국가보안기술연구소
e-mail : jypark@kjist.ac.kr

A Study on Secure Base-Camp for Free-Roaming Agents

Jongyoul Park*, Hong Kook Kim*, Hyung-Geun Oh**, Joong-Gil Park**, Jin-Seok Lee**
*Dept. of Info. & Comm., Kwang-Ju Institute of Science and Technology
**National Security Research Institute

요 약

이동 에이전트는 자율적으로 수행이 가능한 능동적인 객체이지만 이동 에이전트 서버에 의해서 수행되기 때문에 서버에 대한 보안상 취약점을 가진다. 이와 관련하여 다양한 연구가 진행되었고 일정한 조건 안에서 안전한 방법들이 제안되었다. 그 중에는 여러 서버를 중첩하지 않고 순회하는 조건에서 안전한 수행 결과를 보장하는 방법이 있다 [10,11]. 이 방법은 신뢰 서버를 기점으로 에이전트가 자유롭게 순회를 하고 그 결과를 체인화하여 저장하는 방법이다. 본 논문에서 중첩된 순회를 가능하게 하며, 중요한 의사 결정을 안전하게 수행할 수 있는 이동 에이전트를 위한 베이스캠프를 제안한다.

1. 서론

이동 에이전트는 전통적인 클라이언트-서버 모델이 가지는 다양한 문제점을 해결할 수 있는 분산처리 시스템의 한 모델로 많은 주목을 받고 있으며 차세대 통신 구조 및 시스템에서 꼭 필요한 기능으로 평가 되고 있다 [1,3,4].

이동 에이전트란 이동성을 가지는 지능형 에이전트를 의미하는 것으로 지능형 에이전트가 가지는 특징 외에 이동성을 가지고 있어 통신두절의 경우에도 프로그램의 중단 없이 자율적인 수행이 가능하다.

1.1 이동 에이전트와 보안 문제

이동 에이전트는 많은 장점을 가지고 있는 반면에 보안상의 문제점도 가지고 있다. 이 문제점은 이동 에이전트 코드의 수행이 이동 에이전트의 능동적인 판단에 의해서 결정이 되지만 최종적으로는 각 에이전트 서버에 의해서 수행되기 때문에 그 내용이 서버에 노출 된다는 것이다. 코드를 수행하면서 내용을 숨기는 방법들[12,13]도 연구되었지만 아직까지 효과적이지 못하다. 또한 통신 체널을 이용한 방법들도 연구되었지만 이동 에이전트의 특징 중 하나인 **"서버 연결 없이 자율적인 수행"** 이 불가능하다. 반면 에이전트의 코드가 아닌 데이터를 보호가 위한 방법으로 해쉬 체인을 이용한 방법들 연구되었다[10,11]. 해쉬 체인을 이용한 방법은 각 에이전트 서버가 수행한 결과들을 서로 체인화 시켜서 데이터의 비밀성, 무결성, 부인방지, 변조(위조) 방지의 특징을 제공하는 것이다.

하지만 이동 에이전트는 **"외부로 드어나는 중요한 의사 결정"**이나 **"중첩되는 순회"**의 경우 악의를 가진 서버에 의해서 의사결정이 변경되거나 일부 수행 결과가 삭제되는 문제점이 있다. 본 논문은 이와 같은 문제를 해결하기 위해서 이동 에이전트가 신뢰할 수 있는 중간 기점을 제공하고 기존의 해쉬

체인을 이용한 방법과 연동하기 위한 방법을 연구 및 제안한다.

2. 해쉬 체인을 이용한 이동 에이전트 보호

해쉬 체인은 다양하게 정의가 가능하지만 간소화 하여 표현하면 다음과 같은 수식으로 정리된다.

"$h_{i+1} = Hash(h_i, Enc(data_i, r_i), r_i)$, $0 \le i \le n$"

- $f = Hash(x)$: 메시지 x 에 해쉬 함수를 적용
- $f = Enc(x)$: 메시지 x 를 홈 서버의 공개키로 암호화
- h_0: 홈 서버가 생성한 난수
- $data_i$: 에이전트 서버에서 수집한 데이터
- r_i: 에이전트 서버가 생성한 난수

여기서 각 에이전트 서버가 선택한 r_i 는 난수 값으로 홈 에이전트 서버가 알고 있거나 암호화 되어 해쉬체인과 함께 전송된다.

각 에이전트 서버는 자신이 제공한 정보를 자신이 직접 암호화고 전송하기 때문에 h_{i+1} 생성 과정에서 변조의 우려는 없다. 하지만 다른 서버의 데이터를 변경(추가 혹은 삭제) 하는 것이 가능 하다면 자신에게 더욱 이득이 되는 방향으로 바꾸고 싶을 것이다.

만약 이동 에이전트가 에이전트 서버 S_i 를 지나 순회 하다가 다시 $S_{i+a}=S_i$ 를 방문한다면 다음과 같은 경우가 발생할 수 있다.

- 정상:

$$h_{i+a+1} = Hash(h_{i+a}, Enc(data_{i+a}, r_{i+a}), r_{i+a})$$

- 변조: $h_{i+a+1} = Hash(h_i, Enc(data_i, r_i), r_i)$

해쉬 체인은 하나의 메시지인 h_i 에서만 연관이 있기 때문에 h_i 의 값을 임의로 이전 값으로 변경해 버리면 S_{i+1} 에서 S_{i+a} 까지 수집한 모든 데이터는 합법적으로 지워지게 된다.

또 다른 공격 방법으로 변조된 데이터를 삽입하는 경우가 가능하다. 다음의 경우를 보자

$$h'_{i+1} = Hash(h_i, Enc(data'_i, r'_i), r'_i)$$
$$h'_{i+2} = Hash(h'_{i+1}, Enc(data'_{i+1}, r'_{i+1}), r'_{i+1})$$
$$\cdots$$
$$h_{i+a+1} = Hash(h'_{i+a}, Enc(data'_{i+a}, r'_{i+a}), r'_{i+a})$$

위의 수식에서 S_i 부터 S_{i+a} 까지는 그림 1 과 같이 하나의 서버가 가상으로 데이터(h', $data'$, r')를 생성한 경우이다. 최악의 경우 $i = 1$, $n = a+1$ 인 경우를 가정하면 홈 에이전트 서버는 에이전트가 수집한 모든 데이터가 한 서버에 의해서 조작되는 경우도 발생할 수 있다.

따라서 이러한 공격으로부터 데이터를 안전하게 보호하기 위해서는 동일한 에이전트 서버를 중첩되게 방문하는 것을 제한해야 하지만 이동 에이전트

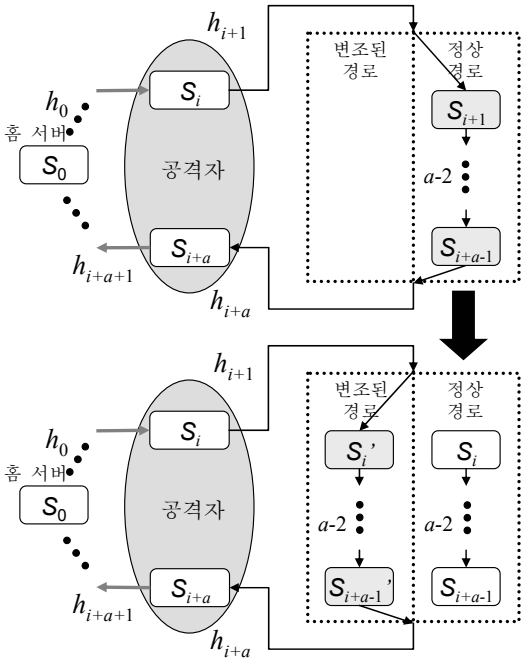

그림 1 변조된 데이터의 생성

의 자율성을 최대한 보장하기 위해서는 중첩되는 방문을 허용해야만 한다. 또한 에이전트가 순회하면서 중요한 의사 결정이 필요한 경우가 생기면 홈 에이전트 서버로 복귀하지 않고 중간에 안전하게 계산이 가능한 중간 기점 또한 필요하다.

조작된 데이터의 생성이 가능한 것은 이동 에이전트가 순회하면서 방문하는 에이전트 서버와 홈 에이전트 서버가 서로 격리될 가능성이 있기 때문이다. 만약 이동 에이전트가 방문하는 에이전트 서버들 사이에 홈 에이전트가 신뢰할 만한 서버가 존재한다면 이와 같은 공격은 불가능해 진다. 3 장에서는 이와 같은 기능을 제공하기 위해서 "안전한 베이스캠프" 라는 신뢰 서버들을 제안하고 이동 에이전트와의 연동하여 어떻게 구성되는지 기술한다.

3. 안전한 베이스캠프와 상세 설계

이동 에이전트를 위한 베이스캠프는 다음과 같이 정의한다.

베이스캠프: "이동 에이전트가 순회, 탐색을 하는 경우 전진기지로서 중요한 의사 결정이나 중간결산을 위한 감사 기능을 수행하는 전진기지이다"

베이스캠프는 이동 에이전트가 순회 도중 다음과 같은 2 가지의 경우 경유하게 된다.

- **중첩된 순회**: 이동 에이전트는 자유로이 순회 도중 이전에 방문 했던 호스트를 다시 방문해야 하는 경우 주변의 베이스캠프를 방문하여 중첩 순회를 요청하고 베이스캠프는 에이전트의 순회 경로에서 중첩되는 에이전트 서버를 기준으로 중간중간에 위치하게 된다. 그림 2 는

그러한 사례를 보이고 있다. 우선 AS_2, AS_9 는 동일한 서버이며 "AS_2→ AS_3 → … AS_9 → AS_{10}" 의 수행 경로는 "AS_2 → BC_1 → AS_3 → … AS_9 → BC_1 → AS_{10}"로 변경된다. 이는 에이전트가 중첩된 순회를 결정하면 첫번째 방문 이후부터 다음 방문 이전에 베이스캠프를 방문하고 두번째 방문 이후에 다시 방문 하는 것이다.

- **중요한 의사결정**: 이동 에이전트가 수집한 데이터를 이용하여 중요한 연산(물건 구매 결정, 에이전트의 역할 변경, 중요 데이터의 전송 등등)의 경우 베이스캠프로 이동하여 수행한다.

3.1 안전한 베이스캠프

베이스캠프를 구성하기 위해서는 다음과 같은 특징을 가지는 신뢰 그룹이 필요 하다.

- 이동 에이전트는 쉽게 베이스캠프를 찾아야 한다.
- 이동 에이전트의 암호화된 코드는 베이스캠프만이 복호화 할 수 있고 수행할 수 있다.
- 이동 에이전트의 순회는 베이스캠프를 기준으로 한다.

베이스캠프는 네트워크 관리자 혹은 전체 시스템에서 정의된 신뢰 서버들로 구성이 되며 각각의 서버들은 전체 신뢰그룹의 비밀키를 공유한다. 반면 각 이동 에이전트 및 홈 에이전트 서버는 신뢰그룹의 공개키를 공유한다. 전체 신뢰 그룹과 이동 에이전트는 비대칭적인 키를 보유함으로써 합법적인 베이스캠프를 쉽게 이용할 수 있고, 베이스캠프는 에이전트가 신뢰그룹의 공개키를 이용하여 쉽게 인증할 수 있다. 그룹키 관리를 위한 시스템은 [14]에서 자세히 기술하고 있으며 본 논문의 논지에서 벗어나므로 기술을 생략하고 이동 에이전트는 신뢰그룹의 공개키를 신뢰서버는 신뢰그룹의 비밀키를 알고 있다고 가정한다.

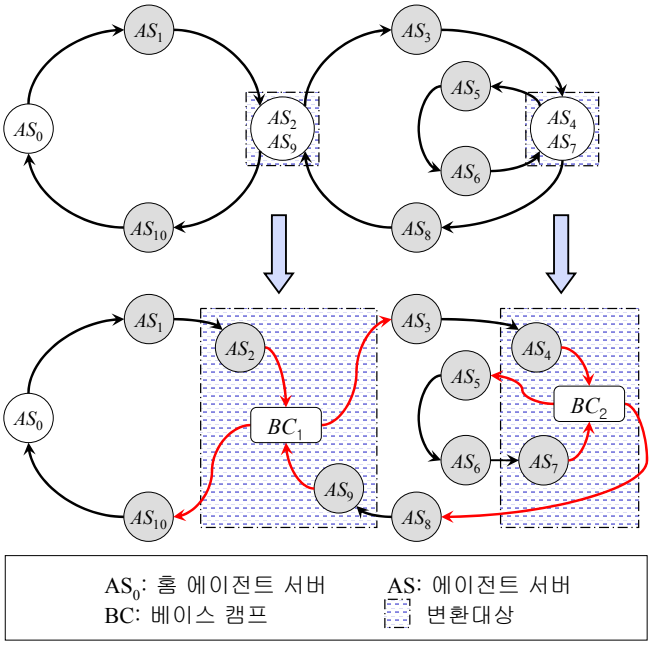

그림 2 베이스캠프의 적용 사례

3.2 시스템 구성

본 논문에서 이동 에이전트 코드는 두가지 형태로 분리한다. 각각의 특징은 다음과 같다.

- **일반 코드**: 이동 에이전트의 바탕이 되는 코드로 각 에이전트 서버에서 수행되는 코드를 수반한다. 정보 수집 및 이동을 위주로 한다.
- **암호화된 코드**: 홈 에이전트 서버 혹은 베이스캠프에서 암호화한 코드로 베이스캠프에서만 수행되는 코드로 의사결정과 같은 중요한 코드를 내포하고 있다.

암호화된 코드는 기존의 에이전트 코드와 달리 에이전트 자체가 아닌 에이전트의 일부 데이터의 형태로 저장이 되며 신뢰그룹의 공개키로 암호화 되고 홈 에이전트 서버의 서명을 포함하여 전송된다. 결과적으로 베이스캠프를 제외한 다른 에이전트 서버는 암호화된 코드를 읽거나 변경할 수 없고 수행할 수도 없다. 이것을 베이스캠프 기준으로 정리하면 다음과 같다.

- $AS_0 \rightarrow BC_1 : Enc_g(MC, Sign_{AS_0}(MC), r_1), r_1)$
- $BC_1 \rightarrow AS_3 : Enc_g(MC, Sign_{BC_1}(MC), r_2), r_2)$

$y = Enc_g(x)$: 메시지 x 를 신뢰그룹의 공개키로 암호화한 값
$y = Sign_k(x)$: 메시지 x 에 대한 k 의 전자서명 값
A → B: x : 메시지 x 를 A 에서 B 로 전송

AS_n: 에이전트 서버 — S_0: 홈 에이전트 서버
BC_1: 베이스캠프 — MC: 수행할 명령어(코드)
r_1, r_2: 난수

베이스캠프는 홈 에이전트 서버로부터 수신한 원본 이동코드(MC)를 수행하고 자신의 서명과 함께 다음 호스트에 전송한다. 이후 호스트부터는 BC_1 을 홈 에이전트 서버로 인지하고 이동 에이전트를 수행한다. 모든 수행을 마치고 다시 BC_1 으로 복귀한 이동 에이전트는 수행결과를 기반으로 필요한 작업을 수행하고 정리하여 결과를 저장한다. 여기서 저장되는 결과물은 베이스캠프 이후에 방문한 정보를 포함하지만 각각의 에이전트 서버의 값으로 저장되지 않고 베이스캠프의 정보로 같이 저장된다. 홈 에이전트 서버의 입장에서는 베이스캠프 BC_1 이후의 정보는 BC_1 이 제공한 것으로 인지한다는 것이다.

3.3 해쉬 체인과 베이스캠프

해쉬 체인을 이용한 방법은 각 에이전트 서버가 제공하는 정보를 각 호스트에서 체인화하여 저장하는 방식으로 이동 에이전트가 방문한 에이전트 서버 하나 하나가 모두 연결되어 있다. 본 논문에서 제안한 베이스캠프를 이용하면 이와 같은 체인화된 연결고리는 베이스캠프 기준으로 끊어지고 다시 정리해야 한다. 이렇게 정리된 내용은 홈 에이전트 서버에게는 하나의 신뢰 서버에서 수행된 것과 같게 취급된다. 즉 내부적으로 이동 에이전트 코드는 2 가지 이상으로 이분화되어야 하고 일부는 암호화 되어서 상황에 맞게

수행 되어야 한다. 다행히 이러한 기술은 Java 1.1 의 ClassLoader 를 응용하면 구현이 어렵지 않으며 그림 4 의 오른쪽 상단에 구현된 결과 화면을 볼 수 있다.

4. 구현 및 응용

베이스캠프를 구현하기 위해서는 다음과 같은 두가지를 구축해야 한다.

- **그룹 키 관리**: 베이스캠프는 해당되는 에이전트 서버는 신뢰할 수 있는 서버라는 가정에서 출발하고 있다. 분산 시스템의 환경 혹은 차세대 통신 환경에서는 광범위하게 분산되어 있는 호스트들 사이에서 신뢰서버를 관리 해야 하기 때문에 대규모의 신뢰그룹 관리 기술이 필요하다. 따라서 트리 구조를 가진 그룹 키 구조를 기반으로 구축해야 한다.
- **이동 에이전트 코드의 이분화 및 암호화**: 이동 에이전트의 코드는 메인 코드와 암호화 되는 코드로 분리된다. 그림 4 는 이러한 관계를 그림으로 그리고 있다. 암호화되는 코드는 이동 에이전트의 코드가 아닌 코드에 내장되는 데이터의 형식으로 저장된다. 즉 경우에 따라 수행해야 하는 각종 코드 집합(class 집합)을 각각 직렬화(Serialization)하고 암호화 하여 이동 에이전트의 데이터로 저장이 된다. 이 데이터는 일반적인 에이전트의 수행시에는 사용되지 않고 베이스캠프에 도착하는 경우에만 복호화되어 ClassLoader 에 의해서 별도 수행이 된다. 다만 이 코드의 수행을 요청하는 쪽이 이동 에이전트의 코드이기 때문에 서로 연동은 어렵지 않다.

이동 에이전트 코드를 이분화 하여 서로 다른 용도로 사용하는 기법은 이동 에이전트가 수행하는 서버들의 등급을 나누고 서로 다른 등급으로 프로그램을 수행하는데 목적이 있다. 만약 에이전트 서버 그룹을 더욱 다양한 보안 등급으로 나눌 수 있다면 네트워크 환경에서의 접근 제어의 권한을 부여할 수 있다.

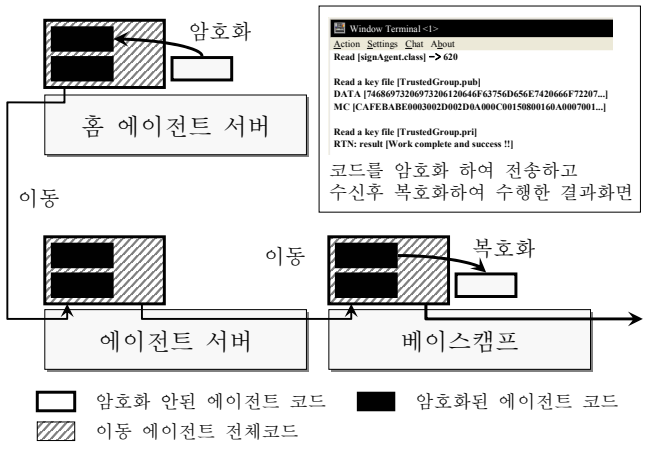

그림 3 에이전트 코드의 일부를 암호화 하여 수행

5. 결론

본 논문에서는 이동 에이전트가 순회 도중 "중대한 의사 결정" 혹은 중첩되는 순회의 중간에서 중간 기지로서 안전한 베이스캠프를 제시하고 구현 및 기존의 이동 에이전트 시스템과 연동에 대한 방법을 제시하고 있다. 특히 해쉬 체인을 이용한 이동 에이전트의 보호 방법에서 문제점으로 제시되고 있는 중첩되는 서버의 방문 허용과 중간 연산이 가능해 지기 때문에 시스템 구성이나 이동 에이전트의 자율성을 최대한 지원할 수 있게 된다.

참고문헌

[1] M. Weiser, "The Comptuer for the 21st Century," Sci. Amer., Sept., 1991.
[2] NIST, http://www.nist.gov/smartspace/downloads/pc2000/sld002.htm, 2000.
[3] R. K. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, H. Yang, "The Case for Cyber Foraging," In Proceedings of the 10th ACM SIGOPS European workshop, Saint-Emilion, France, September 2002.
[4] R. Campbell, D. Sturman, T. Tock, "Mobile Computing, Security and Delegation," the International Workshop on Multi-Dimensional Mobile Communications, 1994.
[5] Ed Brinksma, H. Hermanns, J. Katoen, "Lectures on Formal Methods and Performance Analysis," First EEF/Euro Summer School on Trends Computer Science, Netherlands, Springer-Verlag, LNCS 2090, 2001.
[6] Mignotte, M., "How to share a secret?," Cryptography – Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, Springer-Verlag, Berlin, pp. 371, 1983.
[7] C. Wong, M. Gouda, S. Lam, "Secure group communications using key graphs," IEEE/ACM Transactions on Networking, Volume. 8, Issue 1, p. 16-30, Feb. 2000.
[8] M. Baugher, R Canetti, L. Dondeti, F. Lindholm, "MSEC Group Key Management Architecture," IETF Internet Draft, Sept., 2003.
[9] W. Farmer, J. Guttman, V. Swarup, "security for mobile agents: Issues and requirements," National Information Systems Security Conference, National Institute of Standards and Technology, 1996.
[10] N. Asokan, C. Gulcu, G. Karjoth, "Protecting the Computation Results of Free-roaming Agents," Mobile Agents 98, LNCS 1477, 1998.
[11] J. Park, D. Lee, H. Lee, "Data Protection in Mobile Agents; one-time key based approach," IEEE ISADS 01, pp. 411-418, 2001.
[12] F. Hohl, "Time Limited BlackBox Security: Protecting Mobile Agents from Malicious Hosts," Mobile Agent and Security, LNCS, pp. 99-113, 1998.
[13] T. Sander, C. Tschudin, "Towards Mobile Cryptography," the 1998 IEEE Symposium on Security and Privacy, 1998.
[14] 박종열, 이동익, 홍순좌, 박중길, 이진석, "유비쿼터스 컴퓨팅을 위한 신뢰그룹 관리," 2003 년도 추계 정보과학회 논문집.