분산 환경에 적합한 확장성있는 안전한 RFID 프라이버시 보호 시스템 설계

신명숙°, 박영옥, 이호영*, 홍성표, 민혜란, 이준
조선대학교 컴퓨터공학과
*초당대학교 정보통신과
msshin@chosun.ac.kr

Design of Secure RFID Privacy Protection System for Distributed Environment

Myeong-Sook Shin°, Ho-Young Lee*, Seong-Pyo Hong, Hye-Ran Min, Joon Lee
Dept. of Computer Engineering School, Chosun University
*Dept. of Information Communication, Chodang University

요 약

최근 유비쿼터스 환경의 실현을 위한 핵심기술로서 RFID 시스템에 대한 연구가 활발히 진행되고 있다. 그러나 RFID 시스템이 가지고 있는 특성으로 인하여 사용자 프라이버시 침해 문제가 대두되고 있으며 이를 해결하기 위한 방법들이 개발되고 있다. 기존의 해시 체인 기법은 프라이버시를 침해하는 공격들에 대해서 가장 안전한 기법이다. 그러나 태그를 식별하기 위해서 백엔드 시스템에서의 계산량이 많다는 문제점이 있다. 따라서 본 논문에서는 이러한 확장성 문제를 해결하기 위해 해시 체인 기법 기반으로 Hellman's method를 적용하여 병행 가능한 부분을 추출한 후 각 노드별로 분할하여 적용함으로써 노드별로 수행하는 방법을 설계한다.

1. 서 론

차세대 인터넷을 기반으로 구축되고 있는 그리드는 지리학적으로 분산되어 있는 고성능, 대용량의 컴퓨팅 자원을 네트워크로 상호 연동하여 원격으로 사용하여 단일시스템처럼 사용할 수 있는 환경이다. 현재 사용되고 있는 RFID[1]는 RFID 시스템이 가지고 있는 특성으로 인하여 사용자 프라이버시 문제[2]를 발생시킨다. 이러한 문제[2] 해결을 위하기 위해서는 급격히 늘어난 태그의 개수에 대해 적절한 시간 안에 태그 식별 작업을 완수할 수 있어야 한다. 일반적으로 프라이버시 보호 기법을 설계하는 데 있어서, 안전성을 높여주기 위해서는 백엔드 시스템에서 처리해야 하는 계산량을 더욱 많이 늘려 주어야 한다. 그러나 백엔드 시스템의 계산량이 어느 정도 이상 많아지게 되면 태그를 실시간으로 식별하는 것이 불가능해지기 때문에, 프라이버시 보호 기법의 연구에 있어서 백엔드 시스템의 성능 측면에 대한 고려가 반드시 이루어져야만 한다. 따라서 컴퓨터 및 네트워크 성능이 향상됨에 따라 이러한 고성능 연산 능력을 요구하는 문제를 해결하는 방법에 초점을 모으고 있다.

기존 연구들은 프라이버시 보호 기법의 연구에 있어서 백엔드 시스템의 성능 측면에 대한 문제를 해결하기 위한 논문이 발표되었다. 그러나 대부분의 방법이 이질적인 시스템과 다양한 네트워크로 연결되는 환경에는 적합하지 않다.

본 논문에서는 RFID 프라이버시 보호를 위해 백엔드 시스템에서의 대규모의 저장 공간과 막대한 계산능력이 필요하다. 이와 같이 태그 식별 작업을 위해서 막대한 계산을 요구할 때 고성능 컴퓨팅 자원을 효율적으로 이용하기 위한 방법으로써 해시 체인 기법[3] 기반으로 Hellman's method 알고리즘[4]을 적용하여 병행 가능한 부분을 추출하고 분할한 후 노드별로 수행하는 방법을 설계한다[5].

2. 관련연구

본 장에서는 확장성을 위한 안전한 프라이버시 보호 시스템에 적합한 연구들을 소개한다.

2.1 해시 체인 기법

M. Ohkubo 등이 제안한 기법[3]으로 그림 2-1과 같이 일방향 해시 함수를 사용하여 안전한 프라이버시 보호가 보장되는 기법이다.

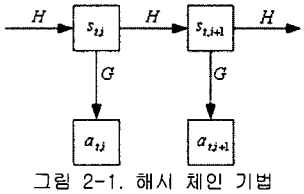

그림 2-1. 해시 체인 기법

백엔드 시스템에는 ID_t와 해시 시드 값 $s_{t,1}$이 저장되며 태그에도 동일한 $s_{t,1}$값을 저장하고, 두 개의 해시함수 H와 G로 구현한다. 리더의 질의에 대해 태그는 $a_{t,i} = G(s_{t,i})$를 수행하여 리더에게 응답하며 자신의 비밀 값인 $s_{t,i}$는 $H(s_{t,i})$를 통해 $s_{t,i+1}$ 로 갱신한다.

그러나 이 기법에서의 문제점은 서버에서 태그를 식별하기 위한 계산량이 많다는 것이다.

2.2 Hellman's method

Hellman에 의해 제안된 방법[4]으로 DES 알고리즘을 적용하여 미리 선택한 평문에 대하여 암호문을 계산하고 계산된 키를 적은 메모리에 저장하는 방법이다.

선택된 평문 P 를 키 S_1을 이용하여 DES로 암호화하고 n 비트의 블록 암호문을 k 비트로 변환하는 함수를 f 함수로 표기하면 f 함수는 식 (1)과 같이 정의된다. 여기서 S_1을 SP(Start Point), S_n을 EP(End Point)라 한다.

$$g : \{0,1\}^n -> \{0,1\}^k$$
$$f(S_1) = g(E_{S_1}(P)) \qquad (1)$$

키 S_1과 f 함수의 관계를 식으로 나타내면 식 (2)와 같다.

$$S_i = f(S_{i-1}) = f^i(S_1), i = 1,2...,n \qquad (2)$$

m개의 키를 갖는 경우 $m^{2/3}$의 메모리와 $m^{2/3}$의 연산으로 키를 찾을 수 있다. Hellman's method는 크게 선행계산 과정과 키 탐색 과정으로 나뉜다.

선행계산 과정은 그림 2-2와 같이 r개의 다른 g함수에 따라 m개의 SP를 n 번 반복하여 EP를 생성하고 생성된 EP에 대하여 정렬한 후 (SP,EP)를 저장하여 g 함수에 대한 테이블을 만든다. 또한 동일한 m 개의 SP에 다른 g 함수에 따른 테이블을 만든다.

```
for q = 1 to r
  for t = 1 to m
    SP_t^(q) 생성
    S_1 = SP_t^(q)
    for i = 1 to n
      S_i = f_s^i(E_{S_{i-1}}(P))
    next i
    EP_t^(q) = S_n
  next t
next q
```

그림 2-2. 선행계산 알고리즘

키 탐색 과정은 암호문(C=E(P, Key))의 Key를 찾는 단계로서 먼저, 변환 함수 g 함수를 적용하여 $Y_1 = g(C)$를 계산하고 g 함수 테이블에서 $t \in \{1,..,m\}$인 EP_t와 Y_1를 비교하여 같으면 SP_t에서 f 함수를 적용하여 $n-1$ 번 계산된 $S_{t,n-1}$를 찾는다. 찾아진 $S_{t,n-1}^{(q)}$의 키로 선택된 평문 P를 암호화하여 풀고자 하는 암호문(C)과 같으면 $S_{t,n-1}^{(q)}$는 찾고자 하는 키이다. EP_t와 Y_1가 같지 않으면 $Y_2 = g(Y_1)$을 계산하고 g 함수 테이블에서 $t \in \{1,...,m\}$인 EP_t 와 Y_2를 비교하여 같으면 SP_t에서 f 함수를 적용하여 $n-2$ 번 계산된 $S_{t,n-2}$를 찾는다. 이러한 과정을 $n-1$ 동안 반복한다.

따라서, 기존 연구들 중에서 프라이버시 보호를 위해서 가장 안전한 방법은 해시 체인 기법이다. 그러나 해시 체인 기법을 적용했을 경우 백엔드 서버에서 태그를 식별하기 위한 계산량이 매우 많기 때문에 확장성을 개선하는 기법이 필요하다. 다음 장에서는 Hellman's method을 적용하여 확장성을 개선하는 제안 기법에 대해서 설명하고자 한다.

3. 제안 기법

기존의 해시 체인 기법은 안전한 보안성이 보장되지만 분산 환경에서 엄청난 태그 수의 증가로 인해 막대한 계산 능력을 요구하는 문제점이 있다. 이러한 문제점을 해결하기위해서 Hellman's method 알고리즘을 이용하여 병행성을 추출한 후 분할한 후 노드별로 수행하는 방법을 설계하여 제안한다.

3.1 병행성 분석

RFID 프라이버시 보호를 위해 적용한 해시 체인 기법은 그림 3-1과 같이 하나의 태그를 식별하기 위한 계산에서, 백엔드 시스템에서는 모든 $1 \le t \le m$와 $1 \le i \le n$에 대해서 $a'_{t,i} = G(H^{i-1}(s_{t,1}))$를 계산한다.

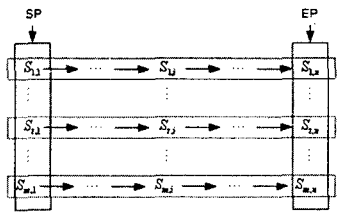

그림 3-1 백엔드 시스템의 해시 체인들

이에 Hellman's method 알고리즘을 적용할 경우 선행계산 과정에서 SP로부터 EP를 계산하는 과정은 서로 다른 SP에 대해 독립적이다. 즉 서로 다른 SP로부터 EP를 계산하는 과정에서 종속성이 전혀 없으므로 이 과정은 동시에 수행될 수 있다. 또한 EP에 대해 정렬하는

과정과 키 탐색 과정도 서로 다른 (SP, EP) 쌍에 대하여 서로 종속성이 없으므로 병행성이 가능하다.

3.2 노드의 분할

계산 그리드를 이용하여 문제를 해결하기 위해 소요되는 시간을 단축하기 위해서는 제기된 문제의 병행성을 충분히 분석하여 동시에 수행될 수 있는 작업으로 분할할 수 있어야 한다. 동시에 수행될 수 있는 작업의 수가 많을수록 그리드의 활용도가 높아진다.

Hellman's method는 임의의 SP를 선택하여 EP를 계산한 후 EP에 대해 정렬 후 키 탐색을 노드별로 수행한다. Hellman's method를 적용한 작업분할 방법은 그림 3-2와 같다.

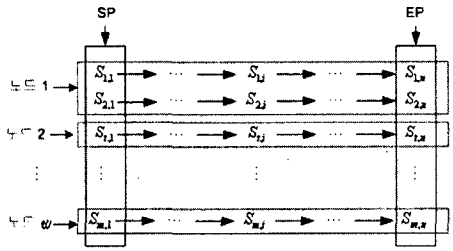

그림 3-2. Hellman's method 작업 분할 방법

위와 같이 작업을 독립적으로 수행될 수 있는 w개의 노드로 작업을 분할하여 그리드 환경에서 동시에 수행할 수 있다면 이상적인 경우 작업 완료시간은 $1/w$로 줄어들 것이다.

3.3 노드 수행 설계

노드들을 수행할 때 기반 기법인 해시 체인 기법과 Hellman's method를 적용하며, 역할에 따라 Master와 Slave로 나눈다.

Salve는 Hellman's method 알고리즘을 수행하여 키를 찾아내는 역할을 한다. Master는 Slave에게 수행할 작업을 할당하고 slave로부터 생성된 결과를 수집하는 역할을 한다. 즉 Master는 Slave를 관리하는 역할을 한다. 노드를 수행하는 Master와 Slave의 역할은 그림 3-3과 같다.

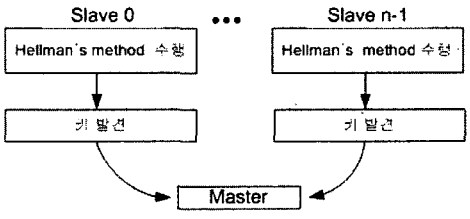

그림 3-3 각 노드에서 수행

위에서와 같이 백엔드 시스템에서의 수행시간을 단축하기 위해 작업을 분할하여 각 노드별로 동시에 수행함으로써 계산 시간을 단축할 수 있는 방법을 설계하였다.

4. 결 론

본 논문에서는 해시 체인 기법과 Hellman's method 알고리즘을 기반으로 병행성을 추출한 후 분할하여 각 노드별로 수행하는 방법을 설계하였다.

RFID 시스템에서 프라이버시 침해 문제를 해결하기 위한 기존의 기법들 가운데 프라이버시 보호를 위해 보안 요건을 모두 보장하는 안전한 기법은 해시 체인 기법이다. 그러나 이 기법은 백엔드 시스템의 확장성 측면에서 문제가 있기 때문에 실제로 활용되기에는 어려움이 있다. 본 논문에서는 이러한 확장성 문제를 해결하기 위해서 먼저 해시 체인 기법에서 해시 체인들이 서로 독립적으로 계산이 가능하다는 점을 이용하였으며, Hellman's method 알고리즘 적용에서 키를 알아내는 과정 중 서로 다른 SP에 대해서 전혀 종속성이 존재하지 않는 점을 이용하였다. 따라서 각 노드에서 선택된 임의의 SP를 이용하여 (SP, EP) 쌍을 계산하는 과정과 키 탐색 과정을 독립적으로 수행한다는 것을 이용하여 병행성을 추출한 후 각 노드별로 분할하였다.

따라서 본 논문에서는 안전한 프라이버시를 위해서 해시 체인 기법을 적용하였으며, 확장성을 위해서 Hellman's method 알고리즘을 적용함으로써 병행 가능한 부분을 추출하고 분할한 후 노드별로 수행하는 방법을 설계하여 제안하였다. 향후 그리드로 이식하여 구현하는 연구를 진행해 나갈 생각이다.

참고문헌

[1] Willam P. Walsh, Research and application of RFID Technology to enhance aviation security, IEEE, 2000.

[2] 이승구, 여상수, 조정직, 김성권, "RFID 시스템에서 안전하고 효율적인 프라이버시 보호 기법",한국컴퓨터종합학술대회, Vol 32, No 1(A), 2005년.

[3] M. Ohkubo, K. Suzuki, and S. Kinoshita. "Cryptographic approach to "privacy-friendly" tags". In RFID Privacy Workshop, MIT, USA, 2003.

[4] M. Hellman. "A cryptanalytic time-memory trade off". IEEE Transactions on Information Theory, IT-26(4):401:406, 1980.

[5] Hyung-Jun Kim, Sung-up Jo, Yong-won Kwon, So-Hyun Ryu, Yong-je Woo, Chang-Sung Jeong, and hyoungwoo Park, "Fast Parallel Algorithm for Volume Rendering and Its Experiment on Computational Grid", ICCS 2003, INCS 2657, pp.610-618, 2003