계산 그리드를 이용한 대량의 RFID 태그 판별 시간 단축 방법

신명숙* · 이준**

Method to Reduce the Time when Identifying RFID Tag by using Computational Grid

Myeong-sook Shin* · Joon Lee**

요 약

유비쿼터스 시대를 견인해나갈 핵심 기술인 RFID가 전 세계적으로 주목 받고 있다. RFID는 향후 관련 산업의 활성화와 고용창출 및 사회적 투명성을 향상시킬 수 있는 기술이지만, RFID가 지니고 있는 기술적 특성으로 인한 보안 및 프라이버시 등의 문제를 가지고 있어서 RFID의 실용화가 지연되고 있는 것이 현실이다. 본 연구는 RFID 시스템 도입으로 발생 가능한 프라이버시 침해 가능성에 대한 해결방안을 찾고자 기술적인 방법을 도입한다. 도입 결과는 먼저 RFID 태그에 M. Ohkubo 등이 제안한 Hash-Chain 기법을 적용하였는데 태그 수의 증가할수록 막대한 계산 시간을 요구하는 문제점이 발생한다. 이를 해결하기 위하여 SP들을 균등하게 분할하여 그리드 환경의 다중 노드에서 동시에 수행함으로써 태그 판별 시간을 단축하는 해결 방안을 제시하고자한다. 이는 RFID 태그의 프라이버시 보호를 유지하면서 RFID 태그의 실시간 처리가 가능해질 것이다.

ABSTRACT

RFID is core technology to lead ubiquitous computing, and attract the notice of the world. It also improves social transparency, creates employment, and invigorates the allied industries. However, The technical characteristic with RFID has some problems with security and privacy. The commercialization of RFID is delayed due to these problems. This paper introduces the technical method to find solutions about an invasion of privacy to be due to introduce RFID system. First, this method applies Hash-Chain proposed by M. Ohkubo and some other researchers. The more tags increase, the more it demands lots of computation time. We divide SPs equally to solve these problems. And then, We'll suggest solutions to shorten the identification time of tag by implementing SPs with multi nodes of Grid environment at the same time. This makes it possible to keep the privacy protection of RFID tag, and process RFID tag in real time at the same time.

키워드

RFID, 계산 그리드, MPICH-G2, 개인정보보호, 해시 체인 기법

* 조선대학교 전자정보공과대학 컴퓨터공학부(msshin@hanafos.com)
** 교신저자 : 조선대학교 전자정보공과대학 컴퓨터공학부 교수(jlee@chosun.ac.kr)
접수일자 : 2010. 08. 16 심사(수정)일자 : 2010. 09. 03 게재확정일자 : 2010. 10. 15

Ⅰ. 서 론

1970년대부터 사용해 온 바코드를 대체할 기술인 RFID(Redio Frequency IDentification)는 미래 유비쿼터스 시대를 이끌어 나갈 핵심 기술로서 주목 받고 있다. RFID는 무선 기술을 이용하여 원격으로 감지 및 인식하여 정보교환을 가능하게 하는 기술이다. 하지만 RFID를 사용하였을 때 개인의 프라이버시 침해라는 심각한 이슈가 제기되고 있다[1][2]. 따라서 급변하는 환경 속에서 기업의 경쟁우위를 차지하기 위한 수단으로 RFID 시스템 활용 시 개인 프라이버시 침해 가능성을 해결하기 위한 방안 마련이 절실한 실정이다.

기존 연구들 중에서 프라이버시 보호를 위해 저가의 태그를 이용한 RFID 시스템 환경에서 가장 안전한 기법으로 M. Ohkubo 등의 Hash-Chain 기법을 이용하였다[3]. 그러나 이 기법은 백엔드 서버에서 태그 ID를 판별하기 위하여 모든 태그에 대한 정보를 가지고 순차적으로 판별 과정을 수행한다. 실제로 태그 수의 증가로 인해 막대한 계산 능력을 요구하는 문제점을 가지고 있다. 여기에서 m*n Hash-Chain 계산 테이블에서 해시 시드 값 $s_{1,1}, s_{2,1}, \cdots s_{m,1}$을 SP(Startpoints)라고 하고 n번 연산을 마친 마지막 값, $s_{1,n}, s_{2,n}, \cdots, s_{m,n}$을 EP(Endpoints)라고 가정한다.

따라서 기존 계산 작업의 한계를 극복하기 위한 대안으로 지리적으로 분산되어 있는 고성능 컴퓨팅 자원을 네트워크로 상호 연동하여 조직과 지역에 관계없이 가상 집합체들의 자원을 공유하여 대량의 태그 처리를 수행할 수 있는 계산 그리드(Computational Grid)의 활용이 필요하다[4].

본 논문에서는 프라이버시 보호를 유지하면서 이질적인 시스템으로 구성되는 그리드 환경에서 Hash-Chain 계산 테이블을 생성하는 프로그램을 작성하고, SP들을 균등하게 분할하는 SP균등분할 알고리즘을 제안하여 구현한다. 이 때 각 노드의 데이터들 중에서 최초의 키를 찾는 시간을 기준으로 한다.

Ⅱ. 관련 연구

본 장에서는 RFID 프라이버시 보호를 위한 계산 그리드 환경의 태그판별처리 모델에 적합한 연구들을 소개한다.

2.1 RFID

RFID는 IC 칩을 내장한 태그에 축적된 정보를 무선 주파수를 이용하여 원격에서 인식하는 방식이다. 또한 자동식별(Identification)의 기능적인 측면에서는 기존 바코드 시스템의 진화된 시스템이다. 바코드와 가장 큰 차이점은 언제, 어디서나, 자동 확인 또는 위치 추적이 가능하여 정보 갱신 및 수정이 가능하다는 점이다. 이러한 특성을 갖는 RFID가 공급 망에 적용됨으로써 얻을 수 있는 가장 큰 장점은 공급망의 전체적인 가시성(Visibility)을 제공해 준다는 것이다[5]. 이렇게 RFID를 도입하여 조달에서 생산, 분배, 최종 소비자까지의 전체 분배 채널을 관리하기 위한 통합 시스템을 구축할 수 있다. 그러나 태그의 유일한 ID를 가지는 RFID 특성은 보안 및 프라이버시 침해 가능성 문제를 제기한다.

2.2 개인 프라이버시

RFID 시스템 도입으로 인한 개인 정보 침해 문제의 심각성은 개인 신상정보의 단순한 수집, 이용보다 개인의 여러 가지 거래 내용, 사회활동 내용과 신상정보를 조합함으로써 본인도 모르는 사이에 분석 활용될 수 있다는 점이다[6]. 결국 기본적으로 개인정보 침해 문제는 정보가 여러 영역에서 수집, 저장, 축척, 관리, 활용되는 것에 기인하며, 개인의 다양한 성향을 사전 동의 없이 제 3자에게 보여주게 되는 것이다.

정부망은 시스템을 이용한 프라이버시 침해 경우를 다음 4가지 경우로 구분한다. ① RFID 태그에 개인정보를 기록하는 경우, ② RFID 태그에 기록된 개인정보를 수집하는 경우, ③ RFID 태그의 물품정보와 특정 개인의 정보를 연계하는 경우, ④ 기타 태그가 부착된 물품을 구매하거나 사용하는 경우 등이다. 이 때 단순히 RFID 태그의 물품 정보를 활용하여 재고관리, 창고관리 등만을 수행하는 경우, 즉 물품정보와 개인정보가 연계되지 않는 경우는 프라이버시 보호 적용에서 제외할 수 있다.

따라서 실제로 RFID 시스템을 적용할 때 개인 프

라이버시가 침해받을 수 있는 영역은 소비자가 태그가 부착된 제품을 구매할 때 나타나게 된다.

2.3 Hash-Chain 기법

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

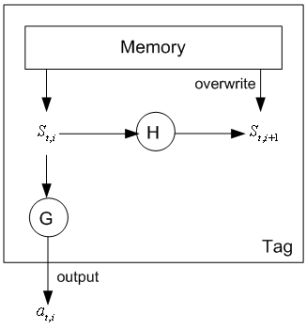

그림 1. Hash-Chain 기법의 태그 내부 구조
Fig. 1 Tag Inner Structure of Hash-Chain Scheme

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 시드 값인 $s_{t,i}$는 $H(s_{t,i})$를 통해 $s_{t,i+1}$로 갱신한다. 그러나 이 기법은 서버에서 태그를 판별하기 위한 계산량이 많다는 문제점이 있다.

2.4 그리드 컴퓨팅

그리드 컴퓨팅이란 지리적으로 분산된 컴퓨터를 결합시켜 발생 장소를 의식하지 않고 원하는 만큼 사용할 수 있게 한다는 것에서 나온 것 이다. 현재 각 국가에서는 실제로 그리드 구축이 진행되고 있지만 아직은 미흡한 단계이다. 그리드 시스템은 계산 그리드, 데이터 그리드, 액세스 그리드로 분류할 수 있다[7].

2.4.1 계산 그리드

많은 자원을 연결하여 계산을 해결할 수 있게 하고, 분산 슈퍼컴퓨팅 관련 그리드는 많은 컴퓨터를 동시에 병렬로 사용하게 하여 작업의 전체 실행시간을 줄일 수 있다. 예를 들면, 기계 항공 분야나 물리 분야의 초대형 문제들을 해결하기 위해 사용된다. High-Throughput 컴퓨팅은 주어진 시간 안에 가능한 한 많은 작업을 처리하는 것이 목표다.

2.4.2 데이터 그리드

원격지의 분산된 자료들을 통합하여 분석할 수 있게 해 주는 그리드다. 대표적인 응용분야로는 여러 분산된 곳에서 생성된 자료를 한곳에서 처리하는 형태의 데이터 마이닝이나 분산된 대용량의 데이터를 여러 곳에서 처리해야 하는 고에너지 물리분야, 유전자 정보처리 분야 등이 있다.

2.4.3 액세스 그리드

분산처리를 필요로 하는 어플리케이션을 위한 그리드다. 액세스 그리드도 역시 사용분야에 따라 나뉠 수 있는데, On-demand 그리드는 사용자 작업의 요구에 따라 동적으로 자원들을 연결할 수 있는 그리드다. 액세스 그리드는 그리드를 통해 인간의 상호작용(Interaction)을 지원하는 자원의 융합이다. 이것은 멀티미디어 디스플레이, 프레젠테이션, 인터랙션 환경, 그리드 미들웨어의 인터페이스, 가시화 환경의 인터페이스 등으로 구성된다. 액세스 그리드를 통해 대규모의 분산 환경 회의, 협업 회의, 세미나, 강의, 교육 및 훈련 등이 지원된다. 액세스 그리드에서 노드의 개념은 특히 그리드에 “그룹” 액세스를 제공하는 것이 목적이다.

2.5 글로버스 툴킷

글로버스 툴킷은 그림와 같이 그리드 서비스를 제공하는 미들웨어로서 계산 그리드를 구축하는데 필요한 기본적인 기술이다. 이것은 그리드와 그리드 애플리케이션을 지원하는 커뮤니티 기반, 공개 구조, 공개 소스 집합의 서비스 및 소프트웨어 라이브러리로서 보안, 정보 발견, 자원 관리, 자료 관리, 통신 오류감지, 이식성 등 그리드에서 필요한 다양한 서비스들을 독립적인 요소로 제공한다.

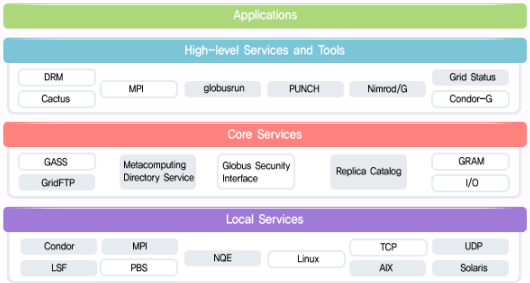

그림 2. 글로버스 툴킷 구조
Fig. 2 Structure of Globus Toolkit

글로버스 툴킷은 크게 그리드 보안, 정보 서비스, 자원 관리, 데이터 관리 등으로 나누어진다. 글로버스 툴킷에서는 GRAM, MDS, GridFTP, GSI 등 핵심 서비스만 제공하고 응용 프로그램이 자신의 목적에 필요한 서비스를 선택/조합함으로써 다양한 형태의 응용프로그램 및 패러다임을 지원한다.

2.6 MPICH-G2

MPICH-G는 MPI에 글로버스가 제공하는 서비스를 가미함으로서 그리드 컴퓨팅에서의 병렬 프로그래밍 환경의 기초를 구성하였으나, 성능이 기존에 사용하던 벤더들의 MPI에 비해 현저하게 떨어지는 단점이 있어 널리 사용되기에는 큰 부족함이 있었다. 그 후 성능의 향상에 초점을 맞추어 지속적인 연구를 거듭하여 MPICH-G2(Grid-enable MPI Chameleon 2)를 개발하였으며 MPICH-G2는 성능면에서 일반 벤더들의 MPI에 뒤떨어지지 않는다. MPICH-G2는 MPICH-G와 마찬가지로 글로버스가 제공하는 많은 서비스를 이용하고 있지만 MPICH-G에서 사용되었던 Nexus를 제거함으로써 많은 성능을 향상 시켰다. Nexus는 여러 개의 프로토콜을 지원하고 자동적인 Data의 변환을 지원하는 등 여러 가지 매력적 기능으로 오랫동안 MPICH-G의 통신 기반 구조로 사용되었지만 그 외 여러 가지 향상시켜야하는 기능들에 대한 문제로 제기 되었다. MPICH-G2는 모든 통신을 직접적으로 다루도록 재 구현 하였다[8].

III. 제안한 방법

기존의 Hash-Chain 기법은 안전한 보안성이 보장되지만 분산 환경에서 엄청난 태그 수의 증가로 인해 막대한 계산 시간을 요구하는 문제점이 있다. 이러한 문제점을 해결하기 위해서 Hash-Chain 기법의 병행성을 분석하여 그리드 환경으로 이식하고, 각 노드별로 SP를 균등하게 분할하여 처리하는 SP균등분할 알고리즘을 제안한다.

3.1 그리드 환경으로의 이식

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

Hash-Chain 기법에서 SP로부터 EP를 계산하는 과정은 그림 3과 같이 서로 다른 SP에 대해 독립적이다. 즉, 서로 다른 SP로부터 EP를 계산하는 과정에서 서로 간섭이나 종속성이 전혀 없으므로 이 과정은 동시에 수행될 수 있다. 또한 태그 판별 과정도 서로 종속성이 없이 독립적이다.

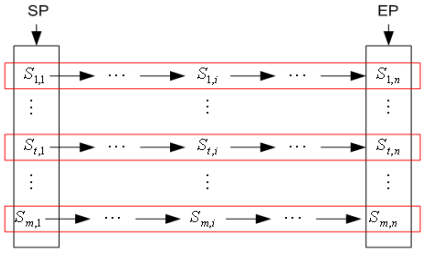

그림 3. 서로 다른 SP에 대해 독립적
Fig. 3 Independence from mutually different SP

3.2 SP균등분할 알고리즘

본 논문에서는 RFID 태그를 판별하는데 막대한 계산 시간을 요구하기 때문에 계산 그리드를 활용하기에 적합하다. 계산 그리드를 이용하여 문제를 해결하기 위해서 소요되는 시간을 단축하기 위해서 먼저 그리드로의 이식 작업이 필요하다.

계산 그리드를 이용하여 문제를 해결하기 위해서는 제기된 문제의 병행성을 분석한 후 k개의 노드로 SP들을 균등하게 분할한다. 이를 식으로 나타내면 식 (1)과 같다.

$$SP[i] = \frac{m}{k} \quad \cdots\cdots (1)$$

여기서 SP[i]는 분할된 SP들의 개수이고, m은 전체 태그 개수이다. 또한 k는 노드 수이며 i는 {1, 2, 3…k}이다. 동시에 수행될 수 있는 노드 수가 많을수록 그리드의 활용도는 높아진다.

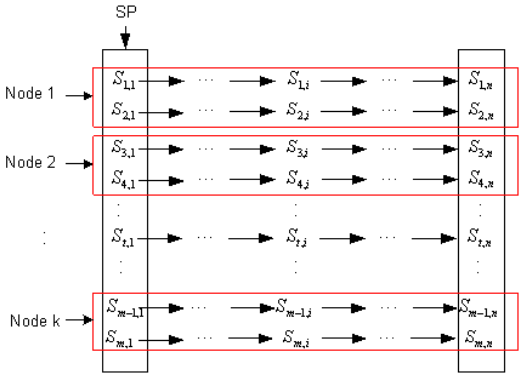

그림 4. 노드별로 균등하게 SP들 분할
Fig. 4 Division of SPs with the Same Size per each Node

그림 4는 Hash-Chain 계산 테이블에서 k개의 노드로 SP들을 균등하게 분할하는 방법을 나타낸다. 이때 각 노드에 균등하게 SP들을 분할할 경우 각 노드에서 선택할 SP들의 개수는 m/k이 된다. 그 이후의 과정은 선택된 SP들을 이용하여 각 노드에서 독립적으로 태그 판별 과정을 수행한다. k 개의 슬레이브에서는 m/k 개의 SP들에 대하여 Hash-Chain을 계산하면서 마스터에서 받은 $a_{t,i}$와 비교하면서 일치하는 값을 찾는다. 여기서의 검색 방식은 순차 검색을 이용한다. 결국 슬레이브는 m/k 개의 SP들에 대하여 Hash-Chain 계산을 하면서 $a_{t,i} = a'_{t,i}$를 비교하여 일치하는 값을 찾게 되면 해시 시드 값 $s_{1,1}, s_{2,1}, \cdots, s_{t,i}, \cdots, s_{m,1}$에 해당하는 식별 정보 ID_t를 마스터에게 전송하고 프로토콜을 종료하게 된다.

위에서 설명한 균등하게 SP들을 분할하는 흐름도는 그림 5와 같다.

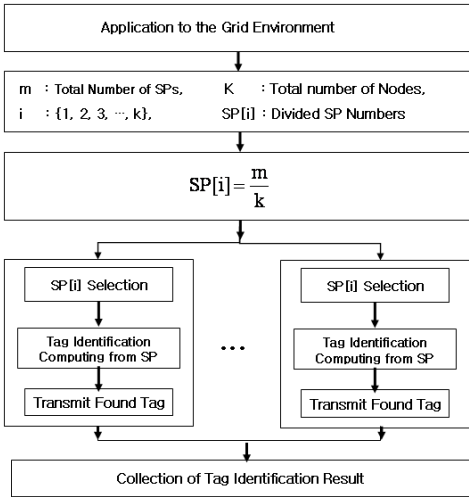

그림 5. SP들의 균등분할을 이용한 흐름도
Fig. 5 Flowchart using Even Division of SPs

기존 방법과 마찬가지로 태그/리더-그리드시스템간의 통신은 무선 전파를 이용한 통신으로써 도청의 가능성이 있는 것으로 간주한다. Hash-Chain 기법에서 단일 서버가 수행하는 해시 연산을 계산 그리드 환경을 구성하는 다중 노드에서 동일한 크기로 SP들을 분할하여 각 노드에서 동시에 수행함으로써 태그 판별 시간을 단축한다.

Ⅳ. 실험 및 분석

4.1 실험 환경

성능평가를 위한 실험 환경은 표1과 같이 구성된다.

표 1. 구현을 위한 하드웨어 및 소프트웨어 환경
Table 1. Environment of Software and Hardware for Implement

항 목		내 용
하드웨어	Master	Intel Xeon 5130 2.0G, dual core 4.0GB
	Slave1	Pentium4 2.4G 512M
	Slave2	Pentium4 3.0G 512M
	Slave3	Pentium4 2.8G 512M
	Slave4	Intel Xeon 5130 2.0G,

		dual core 4.0GB
소프트웨어	운영체제	RadHat Linux 9.0, 커널버전 : 2.4.20-8
	미들웨어	Globus Toolkit 2.2.2
	MPI	MPICH-G2
	언 어	C

여기서 제안 모델은 Master와 Slave로 구성된다. Master 역할을 하는 기계는 Intel Xeon 5130 2.0G, dual core 4.0GB 기계를 사용한다. Master와 Slave는 LAN을 통해 연결되며 Hash-Chain 연산을 하여 태그를 판별하는데 이용한다. 그리고 Hash-Chain 연산에 사용한 일방향 해시 함수는 128bit의 md4, md5를 이용하였으며 검색 방법은 순차 검색을 이용하였다.

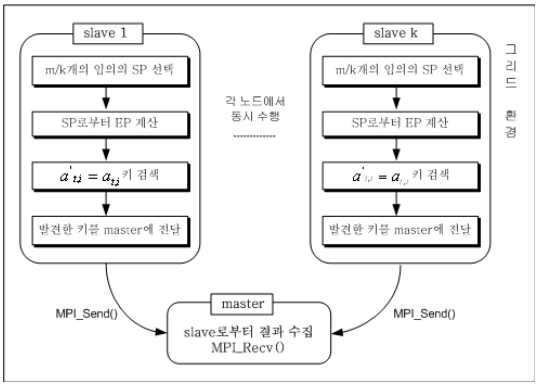

그림 6. Master 와 Slave의 역할 및 통신 방법
Fig. 6 Communication Method and Role of Master and Slave

그리드 환경에서 메시지 전달은 Master와 Slave 사이의 통신을 위해서 필요하다. 본 논문에서는 그림 6과 같이 Master와 Slave 사이의 통신을 위해 MPICH-G2를 이용하였으며 MPI_Recv()와 MPI_Send() 함수를 이용하여 메시지 교환을 한다.

RFID 태그의 샘플 데이터는 16byte의 시드 값을 갖는 임의의 데이터를 생성하여 사용한다. 전체 태그의 개수 m 즉 SP는 1000, 2000, 3000, 4000개로 증가시키면서 실험하였고 최대 Hash-Chain의 길이 n은 1000으로 실험하였다. 성능평가 방법은 Hash-Chain 길이(n)와 태그판별개수는 고정된 상태에서 SP 개수와 노드 수를 증가하면서 태그판별시간을 평가하였다. 단 테스트는 각각 100번씩 수행하여 평균으로 산출하였다.

4.2 보안 분석

기존의 기법들과 제안한 방법에 대해 필수 보안 요건인 기밀성, 불구분성, 전방보안성 등의 보장 여부는 아래와 같이 보장한다.

기밀성은 태그가 송신하는 값은 일방향 해시 함수 G의 계산 결과 값 $a_{t,i}$는 물품에 대한 어떠한 정보도 가지고 있지 않기 때문에 문제가 되지 않는다. 또한 태그의 내부에도 물품에 대한 직접적인 정보가 들어있지 않기 때문에 공격자가 태그의 송신정보를 통해서 물품에 대한 유용한 정보를 획득할 가능성은 없다. 따라서 본 제안 기법에서는 RFID 시스템에서 프라이버스 보호를 위한 필수 요건 중 하나인 기밀성을 만족한다. 불구분성은 공격자가 태그의 다음 단계의 송신 값 $a_{t,i+1}$을 현재까지의 모든 송신 값 $a_{t,i}$를 가지고 예측하기 불가능하다. 이는 일방향 해시 함수 G의 입력 값 $s_{t,i+1}$을 공격자가 알아내기는 어렵기 때문이다. $s_{t,i+1}$을 예측할 수 있으려면 $s_{t,i}$ 값을 $a_{t,i}$알아내야 하는데 $s_{t,i}$ 값은 외부로 전송하지 않고, G 해시 함수를 거쳐 계산된 결과를 전송하기 때문에 일방향 해시 함수의 특성상 $s_{t,i}$를 예측하는 것은 계산적으로 불가능하다. 따라서 제안한 기법에서는 RFID 시스템에서 프라이버시 보호를 위한 필수 요건 중 하나인 불구분성을 만족한다. 전방보안성은 태그가 물리적인 공격을 당하는 경우에는 현재의 시드 값 $s_{t,i}$가 노출이 되지만, 공격자는 $s_{t,j}$ 값을 알아낼 수 없다. 이는 일방향 해시 함수 H의 일방향성을 근거로 한다. 또한, 공격자는 태그가 전송하는 값 $a_{t,i}$를 알아내었다하더라도 G 해시 함수의 일방향성을 근거로 $s_{t,i}$를 알아내기는 어렵다. 따라서 본 제안 기법에서는 RFID 시스템에서 프라이버시 보호를 위한 필수요건 중 하나인 전방보안성을 만족한다.

4.3 실험 결과

본 연구에서 제안한 SP균등분할 알고리즘을 적용

하여 성능을 측정하였다.

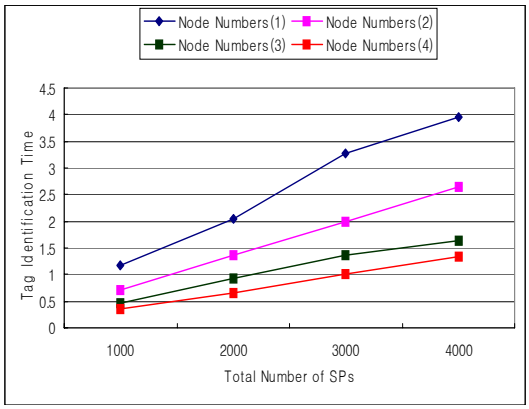

그림 7. SP들의 총수에 따른 성능 비교
Fig. 7 Performance Comparison by the Total Number of SPs

수행 결과는 SP들의 총수에 따른 방법으로 평가를 하였으며, 성능평가 방법은 그림 7과 같이 Hash-Chain 길이 1000은 고정된 상태에서 SP들의 총수를 1000, 2000, 3000, 4000으로 증가하고 노드 수를 1, 2, 3, 4로 확장하면서 SP들의 총수에 따른 태그 판별 시간을 비교하여 평가하였다.

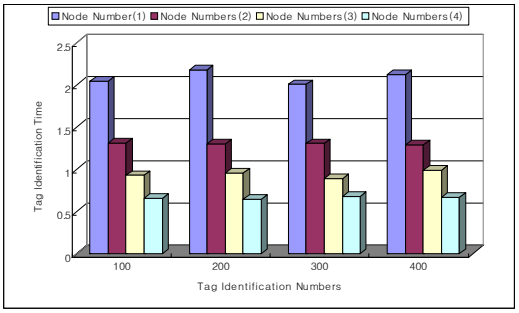

그림 8. 태그 판별 개수에 따른 성능 비교
Fig. 8 Performance Comparison by Tag Identification Numbers

그림 8은 SP들의 총수 2000, Hash-Chain 길이 1000번, 노드 수 1, 2, 3, 4로 고정된 상태에서 태그 판별 개수를 100, 200, 300, 400으로 증가하면서 태그 판별 시간을 비교하였다. 이 때 SP들의 총수와 노드 수가 증가할수록 큰 폭으로 태그 판별 시간이 단축되는 것을 구현 결과를 통해 확인하였다. 또한 태그 판별 개수를 증가하면서 태그 판별 시간을 측정한 결과 태그 수의 증가에도 불구하고 판별 시간의 증감차이가 적음도 보였다.

V. 결 론

본 연구는 RFID 시스템 도입으로 발생 가능한 프라이버시 침해 가능성에 대한 기술적인 해결방안으로써 그리드를 활용하여 태그의 판별 시간이 단축되는 것을 보여주었다.

제안한 방법에서는 기존의 단일 백엔드 서버에서의 작업이 그대로 그리드 환경으로 이관되어 처리되는데, 먼저 태그에 적용한 Hash-Chain 기법의 병행성 분석을 통해 분할된 SP들을 병행처리가 가능하도록 그리드에 이식하였다. 기본적으로 k 개의 노드가 있을 경우, 각 노드에 SP들을 균등하게 분할하는 SP균등분할 알고리즘을 제안하여 구현하였다. 구현을 위한 그리드 환경의 태그 판별처리 모델은 마스터와 슬레이브로 구성된다. 각 시스템의 운영체제는 Ubuntu Linux, 미들웨어는 Globus Toolkit과 MPICH-G2를 탑재하여 사용하였다. 마스터 역할을 하는 머신은 Intel Xeon 5130 2.0G, dual core 4.0GB 머신을 사용하였으며, 슬레이브들은 각각 다른 성능을 가진 머신을 사용하였다. 마스터와 슬레이브 사이의 통신은 LAN을 통해 연결되며 메시지 교환을 위해서 MPI_Recv()와 MPI_Send() 함수를 이용하였다.

구현의 결과, Hash-Chain 길이 1000번, 노드 수 4로 고정한 상태에서 SP들의 총수를 1000, 2000, 3000, 4000으로 증가시키면서 단일 노드와 균등분할을 비교하면 67%, 68%, 72%, 70%의 성능이 향상되었다. 또한 SP들의 총수는 2000 개, Hash-Chain 길이 1000번으로 고정된 상태에서 노드 수 2, 3, 4로 확장하면서 단일 노드와 균등분할 알고리즘을 비교하면 39%, 53%, 68%로 성능이 향상되었음을 알 수 있었다. 테스트는 각각 100번씩 수행하여 가장 먼저 찾은 시간의 합을 100으로 나누어 태그 판별 시간을 산출하였다. 이러한 결과는 많은 계산 능력을 필요로 하는 문제를 그리드 환경을 도입하여 SP들을 균등하게 분할하여 적용한다면 RFID 태그의 프라이버시 보호를 유지하면서 RFID 태그의 실시간 처리가 가능해질 것으로 생각된다.

참고 문헌

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

[2] S. Sarma, S. Weis, D. Engles, "RFID Systems and Security and Privacy Implication," In CHES, vol. 2523 of LNCS, pp. 454-469, August 2002.

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

[4] 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, lNCS 2657, pp. 610-618, 2003.

[5] 이동민, "RFID 기반 상품의 효율적 라이프사이클관리를 위한 통합시스템 설계", 산업공학 학술지, Vol19(4), pp. 333, 2006.

[6] 정국환, "정보사회의 정보이용자 권익보호방안", 한국전산원 연구보고서, 1997.

[7] 이춘희, "그리드 컴퓨팅(Grid Computing)", 정보처리학회논문지, Vol 10(1), 2003.

[8] MPICH-G2, http://www3.niu.edu/mpi

저자 소개

신명숙(Myeong-sook Shin)

1992년 2월 광주대학교 전자계산학과 졸업(공학사)
1996년 2월 광주대학교 대학원 컴퓨터학과 졸업(공학석사)
2008년 8월 조선대학교 대학원 전자정보공과대학 컴퓨터학과 졸업(공학박사)
조선대학교 전자정보공과대학 컴퓨터공학부 겸임교수
※ 주 관심분야 : 시스템소프트웨어, 유비쿼터스컴퓨팅, 정보보호

이준(Joon Lee)

1979년 2월 조선대학교 전자공학과 졸업(공학사)
1981년 2월 조선대학교 대학원 전자공학과 졸업(공학석사)
1997년 2월 숭실대학교 대학원 전자계산학과 졸업(공학박사)
조선대학교 전자정보공과대학 컴퓨터공학부 교수
※ 주 관심분야 : 운영체제, 정보보호, 유비쿼터스컴퓨팅