임베디드 시스템에서 보안 데이터 전송의 전력 사용량 및 오버헤드 분석

이무열 이상헌 진현욱°
건국대학교 컴퓨터공학부
{zlemy, mir1004, jinh°}@konkuk.ac.kr

Analysis of Energy Consumption and Overhead of Secure Data Transfer over an Embedded System

Moo-Yeol Lee Sang-Hun Lee Hyun-Wook Jin°
Department of Computer Science and Engineering, Konkuk University

요 약

대부분의 임베디드 시스템들은 유선 및 무선 네트워크에 연결되어 있으며 이들이 생성하는 데이터는 개인, 기관, 또는 국가에 상당히 중요한 정보일 수 있다. 따라서 통신 시 보안에 대한 요구는 향후 더욱 크게 증가할 것으로 전망된다. 임베디드 시스템에서의 또 다른 중요한 요구사항 중의 하나는 저전력 특성이다. 따라서 중요 데이터의 안전한 전송을 위한 암호화 및 복호화 그리고 네트워크 프로토콜 스택의 수행에 따른 전력 사용량 정보는 차세대 저전력 임베디드 시스템을 설계 및 개발하기 위해서 필수적으로 요구될 것이다. 하지만 기존 연구들은 단편적인 전력 사용량만을 분석하고 있다. 본 논문은 보안 데이터를 위한 암호화, 복호화, 그리고 데이터 송수신을 위한 네트워크 프로토콜 스택의 수행에 걸리는 전력 사용량과 오버헤드를 함께 측정하고 분석한다. 측정 결과 3DES 기반의 암호화 및 복호화 작업 동안의 전력 사용량이 TCP/IP 통신을 위한 전력 사용량 보다 데이터 크기가 증가함에 따라 더욱 크게 측정되었다. 해당 작업의 수행 오버헤드 역시 유사한 성향을 보였다. 그리고 프로세서 코어의 전력 사용량은 시스템 전체 사용량의 10~22% 정도를 차지하는 것으로 측정되었다. 이러한 연구 결과는 임베디드 시스템에서 저전력 보안 데이터 전송을 위해 고려할 사항들을 정량적으로 제시하여 해당 연구 분야에 기여할 수 있다.

1. 서 론

현재 많은 임베디드 시스템들은 이미 유선 또는 무선 네트워크에 연결되어 있으며, 앞으로 더욱 많은 분야에서 널리 사용될 것으로 전망된다. 그에 따라서 이들 임베디드 시스템들이 제공하는 기능의 다양성뿐만 아니라 분야에 따라서는 생산 및 저장하는 데이터의 양과 그 중요성 또한 크게 증가할 것이다. 임베디드 시스템에서 생산되는 데이터들은 개인 데이터에서부터 중요임무를 위한 데이터까지 그 범위는 다양하다. 이들 데이터는 개인, 기관, 또는 국가에 중요한 정보일 수 있으며, 따라서 통신 시 보안에 대한 요구가 증가할 것이다. 이러한 요구를 위해서 데이터 송수신 시 임베디드 시스템에서도 암호화 및 복호화가 동반될 것이다. 현재 Secure Socket Layer (SSL) [1], IPsec [2] 등과 같이 종단 노드 간 또는 네트워크 계층에서의 보안 통신을 위한 프로토콜들이 제안되었으며 일부 사용되고 있다. 하지만 이러한 보안 프로토콜들은 아직 임베디드 시스템에는 그 적용이 미비한 상태이다.

임베디드 시스템에서의 또 다른 중요한 요구사항 중의 하나는 저전력 특성이다. 대부분의 임베디드 시스템은 전지에 의해서 전력을 공급받기 때문에 저전력 특성은 시스템의 수명 및 관리 주기를 결정한다. 따라서 앞에서 언급한 중요 데이터의 안전한 전송을 위한 암호화 및 복호화 그리고 네트워크 프로토콜 스택의 수행에 따른 전력 사용량 정보는 향후 차세대 저전력 임베디드 시스템을 설계 및 개발하기 위해서 필수적으로 요구될 것이다. 하지만 기존 연구들은 단편적인 전력 사용량만을 분석하고 있다. 따라서 위와 같이 일련의 작업들에 대한 거시적인 관점에서의 상세한 전력 사용량 측정 및 분석이 요구된다.

본 연구는 2007년 한국전자통신연구원 위탁과제와 서울시 산학연 협력사업 (CR070019) 지원으로 수행되었음

본 논문은 보안 데이터를 위한 암호화, 복호화, 그리고 데이터 송수신을 위한 네트워크 프로토콜 스택의 수행에 걸리는 전력 사용량을 송신측과 수신측으로 나누어 분석한다. 특히 본 논문에서는 각각의 작업을 세분화하여 각 작업에서 사용된 전력과 수행에 걸린 시간을 함께 측정하고 분석한다. 측정 결과 3DES 기반의 암호화 및 복호화 작업 동안의 전력 사용량이 TCP/IP 통신을 위한 전력 사용량 보다 데이터 크기가 증가함에 따라 더욱 크게 측정되었다. 해당 작업의 수행 오버헤드 역시 유사한 성향을 보였다. 프로세서 코어의 전력 사용량은 시스템 전체 사용량의 10~22% 정도를 차지하는 것으로 측정되었다.

본 논문은 본 서론에 이어 다음과 같이 구성되어 있다. 2장에서는 연구 배경과 관련 연구들을 기술한다. 3장에서는 실험 시스템의 구성과 측정 방법론을 설명하고, 4장에서는 성능 측정 및 분석 결과를 제시한다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

2. 연구 배경

본 장에서는 본 연구의 배경 지식이 되는 TCP/IP 네트워크 프로토콜 스택의 동작과 암호화 알고리즘인 Triple DES에 대해 개괄해서 설명한다. 그리고 본 연구과 관련된 기존 연구에 대해서 토의한다.

2.1 TCP/IP 네트워크 프로토콜 스택

본 절에서는 임베디드 시스템 환경에서 리눅스 커널 버전 2.6에 구현되어 있는 TCP/IP 패킷의 송수신 경로를 설명한다.

송신 프로세스는 송신 시스템 호출(system call)을 통해서 사용자 버퍼에 있는 네트워크 데이터를 커널 버퍼로 복사한다. 시스템 호출은 TCP와 IP 계층을 수행하여 네트워크 데이터를 TCP 세그먼트로 캡슐화(encapsulation)하고 IP 데이터그램으로 캡슐화 한다. 그 후 디바이스 드라이버에 송신 요청을 한다. 이더넷 디바이스 드라이버는 PIO(Programmed I/O)를 사용하여 커널 버퍼에 있는 네트워크 데이터를 이더넷 컨트롤러로 이동시킨다. 이와 같은 일련의 수행이 완료되면 시스템 호출도 성공적으로 반환된다.

수신 노드의 이더넷 컨트롤러에 네트워크 패킷이 수신되면, 인터럽트를 발생시켜서 프로세서가 이를 처리하도록 한다. 디바이스 드라이버에 구현되어 있는 인터럽트 핸들러는 PIO를 사용하여 이더넷 컨트롤러에 수신된 데이터를 커널 버퍼로 이동한다. 그 후 인터럽트 핸들러는 처리할 패킷이 있음을 네트워크 수신 softirq에 알리고 종료한다.

softirq는 리눅스에서 bottom half의 단점을 극복하기 위해서 제안되었다. 리눅스에 구현된 기존 bottom half는 그 종류에 상관없이 하나의 프로세서에서만 수행 가능 하지만, softirq는 같은 종류의 softirq 조차도 여러 프로세서에서 동시 수행 가능하다. 네트워크 수신 softirq는 IP와 TCP 계층의 처리를 수행한다. 따라서 softirq에서 수신된 데이터 패킷에 대한 체크섬 계산, TCP ACK 패킷의 전송, 수신된 ACK 패킷을 기반으로 RTT(Round Trip Time) 계산 등이 수행된다.

커널 버퍼에 수신된 데이터는 사용자 영역의 버퍼로 softirq의 수행 과정 또는 해당 프로세스가 수신 시스템 호출을 수행하는 과정에서 이루어진다. softirq에서 처리중인 패킷의 수신 프로세스가 현재 (current) 프로세스인 경우에는 softirq에서 바로 사용자 버퍼로 복사가 이루어지며, 그렇지 아닌 경우에는 시스템 호출에 의해서 복사가 이루어진다.

2.1 Tripple DES (3DES)

DES는 개인키를 사용하여 데이터를 암호화하는 대칭형 블록 암호화 기법이다. DES는 각 64 비트 데이터 블록에 56 비트 길이의 키(추가 8비트는 에러 검출용)를 적용한다. 이 과정은 여러 가지 모드에서 실행될 수 있으며, 대체와 치환이라는 기본적인 암호화 함수를 반복적으로 16회 반복한다.

DES가 강력한 암호화 기법으로 평가 받고 있지만, 더욱 강력한 암호화를 위해서 Tripple DES (3DES)가 제안되었다. 3DES는 하나의 블록을 암호화할 때 DES 암호화 기법을 세 번 연속해서 적용하는 경우도 있고, 암호화, 복호화, 그리고 암호화 순으로 세 번 적용하여 암호문을 생성하기도 한다. 후자의 경우 복호화 할 때는 복호화, 암호화, 그리고 복호화 순으로 DES 암호화 기법을 적용하여 평문을 생성한다. 각 단계에서 적용되는 키는 서로 다른 두 개의 또는 세 개의 키로 구성될 수 있다.

2.3 관련 연구

최근 데이터 송수신을 위해서 널리 사용되는 TCP/IP의 전력 소비량의 분석이 시작되고 있다 [3][4]. 하지만 이들은 TCP/IP에 대해서만 초점을 맞추고 있으며, 보안 데이터 전송이란 관점에서 측정 및 분석을 수행하고 있지는 않다. 보안 프로토콜에 대한 전력 소비량 분석 [5] 및 전력 소비량 감소 기법들도 [6][7][8][9] 제안되고 있다. 이들은 암호화 및 복호화 또는 키 교환 중에 발생하는 전력 소비량에 초점을 맞추고 있다. 반면 상대적으로 TCP/IP와 같은 통신 프로토콜에 대한 상세한 분석은 생략하고 있다. 본 논문은 암호화 및 복호화와 함께 TCP/IP 프로토콜 스택의 전력 소비량 및 오버헤드를 측정한다. 따라서 기존 연구와 비교해서 보안 데이터 전송을 위한 모든 관련 작업에 대한 전력 소비량에 대한 분석 결과를 제시해 줄 수 있다.

TCP/IP 프로토콜 스택의 오버헤드 및 성능 분석은 많은 연구를 통해서 수행되어 왔다 [10][11][12][13]. 하지만 이들 논문에서 전력 소비량 측정은 병행되지 않았다. 반면 본 논문은 전력 측정 결과와 함께 오버헤드 측정 결과를 제시한다. 이와 같이 본 논문은 성능과 전력 사용량 모두를 보여줌으로써 시스템 설계 및 개발에 더욱 유용한 정보를 제공해줄 수 있다.

3. 전력 측정 방법론

전력 사용량 측정을 위해서 일반적으로 전압계를 사용하는 방법과 전류계를 사용하는 방법으로 그림 1과 같이 구분할 수 있다. 전압계를 사용할 경우에는 그림 1(a)와 같이 회로 상에 저항(R_1)을 삽입하여 수식 1에 의해서 사용 전력을 계산할 수 있다. 전류계를 사용할 경우는 그림 1(b)와 같이 회로에 전류계를 직렬로 연결하고 수식 2에 의해서 사용 전력을 계산할 수 있다. 본 논문에서는 임베디드 보드의 개조가 상대적으로 쉬운 전류계를 사용하는 방법을 사용한다.

$$P = V_2 \times \frac{V_1}{R_1} \qquad \text{(수식 1)}$$

$$P = V_2 \times I_1 \qquad \text{(수식 2)}$$

본 논문에서는 (주)휴인스의 Acumen270s 프로세서 보드[14]를 대상으로 전력 측정을 수행한다. 사용된 임베디드 보드는 Intel XScale PXA270 (520Mhz) 프로세서, 128MB SDRAM, 32MB Flash ROM, 10/100 Mbps Ethernet Controller (LAN91C111)를 장착하고 있으며, 운영체제로는 리눅스(커널 버전 2.6.12)가 설치되어 있다. 본 논문에서는 위의 임베디드 보드 전체에서 사용되는 전력과 프로세서 코어에서 사용되는 전력을 각각 측정하였다. 이를 위해서 그림 2와 같이 두 부분에 전류계를 연결할 수 있도록 보드를 개조하였다.

전류 측정을 위해서는 National Instrument의 PCI-4070 DMM[15]을 사용한다. 해당 DMM은 1.8 MS/s의 측정 주기(10us 주기)까지 지원 가능하며, 이는 세부 기능 단위까지 큰 오차 없이 측정가능한 주기다. 사용된 DMM은 데스크톱의 PCI버스에 연결되며, 데스크톱에서 수행되는 LabView 프로그램에 의해서 측정된 전류 값이 수집된다. 수집된 전류 값은 별도 개발된 프로그램에 의해서 전력(W) 및 구간 별 전력량(J)으로 계산된다.

전류 값을 측정하는 동안 세부 기능별 실행 오버헤드를 측정하기 위해서 본 논문에서는 응용프로그램 수준뿐만 아니라 커널 수준에서 각 기능별 시작 시점과 종료 시점에 타임스탬프를 생성하여 저장한다. 이를 위해서 시스템 타이머를 읽어서 타임스탬프를 생성하고 저장하는 커널 함수를 구현하고, 측정을 위한 각 부분에서 이 함수를 호출하도록 커널을 수정하였다. 타임스탬프의 저장 시 발생할 수 있는 오버헤드와 전력 사용량을 최소로 하기 위해서 /proc 파일 시스템을 사용한다. 그리고 측정이 종료된 후에 /proc 파일 시스템을 통해서 축적된 타임스탬프를 수집하여 분석한다.

현재 본 논문의 측정에서는 DMM에 의한 전력 측정과 타임스탬프에 의한 오버헤드 측정의 시작점 동기를 맞추기 위해서 전력사용량 그래프에 특정 신호를 보일 수 있는 작업 수행과 함께 타임스탬프를 생성토록 하였다. 이를 통해서 전력 사용량 측정 결과와 오버헤드 측정 결과를 함께 분석할 수 있다. 향후에는 GPIO를 사용하여 DMM에 트리거 신호를 보냄으로써 전력 측정 데이터와 오버헤드 측정 데이터 간의 동기를 맞추고 더욱 정확한 분석을 시도하려고 계획하고 있다.

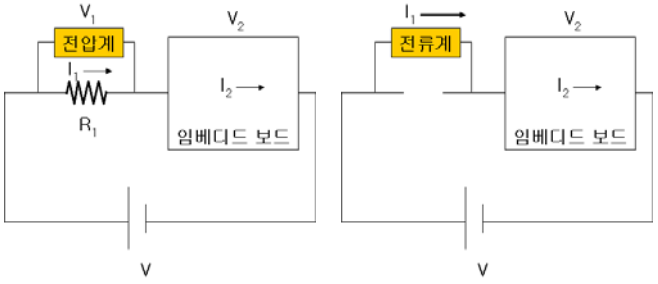

그림 1 (a) 전압계와 (b) 전류계를 사용한 전력 사용량 측정

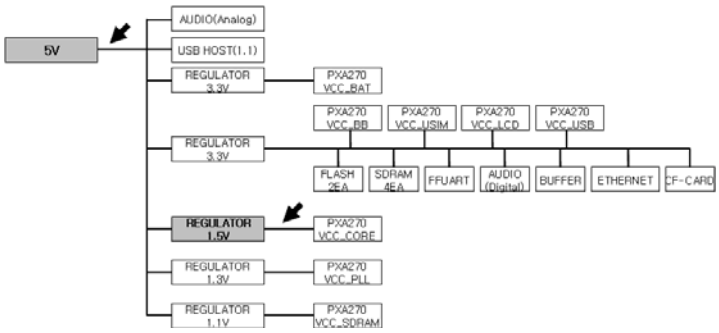

그림 2 Acumen270s 프로세서 보드의 전력 맵과 전류계 연결 위치 (화살표)

4. 전력 사용량 측정 및 분석 결과

본 장에서는 전력 사용량 및 오버헤드 측정 결과를 송신측과 수신측으로 구분하여 설명한다. 송신측에서는 암호화, 시스템 호출, 디바이스 드라이버로 나누어서 측정이 수행되었다. 여기서 시스템 호출 부분은 TCP와 IP 계층을 포함한다. 수신측에서는 복호화, 시스템 호출, softirq, 디바이스 드라이버로 나누어서 측정한다. 수신측에서 TCP와 IP 계층은 softirq에 포함된다. 사용된 암복호화 알고리즘은 3DES이며, 송수신된 데이터의 크기는 512B, 1024B, 1400B이다.

4.1 송신측 전력 사용량 및 오버헤드

그림 3은 송신측에서 프로세서 코어에서 기능별 전력 사용량을 보여주고 있다. 그림에서 보이는 바와 같이 암호화 전력 사용량이 다른 부분에 비해서 크며, 데이터 크기에 따라 급격하게 증가하는 것을 볼 수 있다. 이것은 3DES 알고리즘이 DES 알고리즘을 3번 중복 수행하기 때문이다. 시스템 호출과 디바이스 드라이버에서도 경사는 낮지만 데이터 크기에 따라 전력 사용량이 증가하고 있는 것을 알 수 있다. 시스템 호출은 TCP 계층을 수행할 때 데이터 복사와 체크섬 계산을 수행한다. 따라서 데이터 크기의 증가에 따른 프로세서 코어의 사용률이 증가하고 전력 사용량의 증가를 보인다. 디바이스 드라이버는 이더넷 컨트롤러에 데이터를 전달하기 때문에 데이터 크기에 따른 전력 사용량 증가를 보인다.

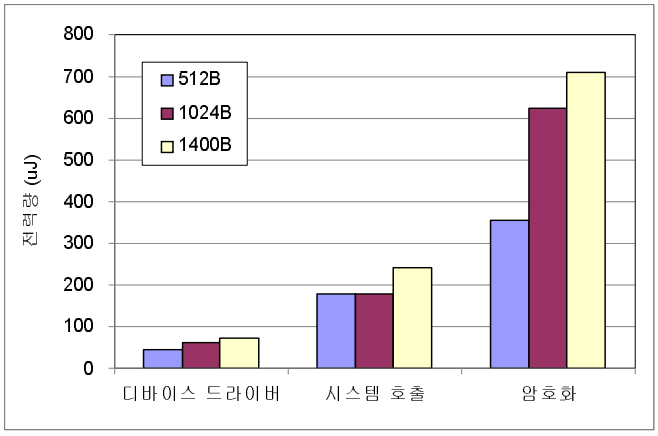

그림 3 송신측 작업별 프로세서 코어 전력 사용량

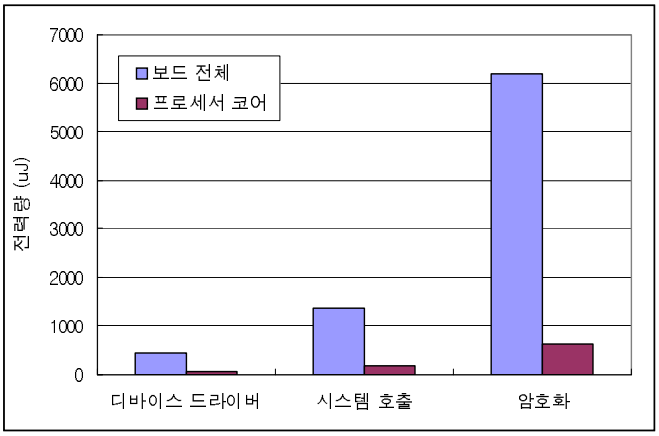

그림 4 송신측 시스템 전체 전력 사용량과 프로세서 코어 전력 사용량 비교 (1024B)

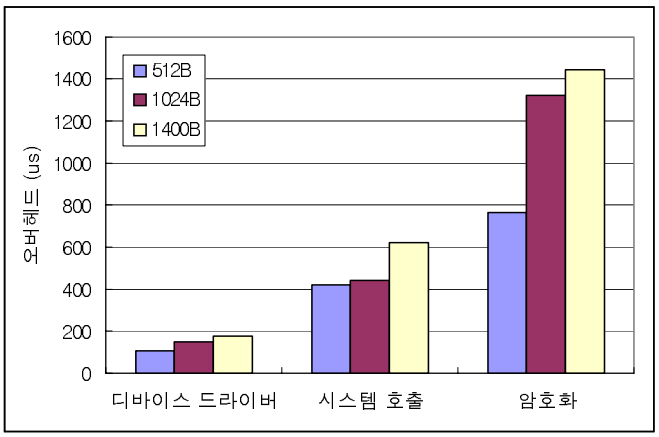

그림 5 송신측 작업별 수행 오버헤드

그림 4는 1024B 데이터에 대해서 암호화와 네트워크 송신을 수행할 때 시스템 전체 전력 사용량과 프로세서 코어에서의 전력 사용량을 비교하고 있다. 그림에서 관찰할 수 있는 바와 같이 시스템 전체와 프로세서 코어의 전력 사용량은 비슷한 성향을 보이고 있으며, 프로세서 코어는 전체 사용 전력량의 10~14% 정도의 전력량을 보이고 있다.

그림 5는 송신측에서 기능별 오버헤드를 보여주고 있다. 그림 5는 그림 3과 아주 흡사한 성향을 보여주고 있다. 이것은 프로세서 코어 전력 사용량이 프로세서 코어에 의한 컴퓨팅 오버헤드에 크게 영향을 받기 때문이다. 디바이스 드라이버에서 데이터 크기에 비례하여 보이는 오버헤드는 데스크톱 환경과는 다른 결과이다. 데스크톱 환경에서는 일반적으로 네트워크 인터페이스 카드에 있는 DMA(Direct Memory Access) 엔진을 사용하여 커널 버퍼에 있는 네트워크 데이터를 네트워크 인터페이스 카드로 이동한다. 따라서 CPU 자원을 절약하고 데이터 크기와 무관하게 낮은 디바이스 드라이버 오버헤드를 보인다. 하지만 임베디드 시스템에서는 2.1절에서 설명된 바와 같이 CPU가 PIO를 통해서 커널 버퍼에 있는 네트워크 데이터를 이더넷 컨트롤러로 직접 이동시킨다. 따라서 데이터 크기에 비례하여 오버헤드가 증가한다.

4.2 수신측 전력 사용량 및 오버헤드

그림 6은 수신측에서 프로세서 코어에서 기능별 전력 사용량을 보여주고 있다. 수신측에서도 데이터 크기의 증가에 따라 복호화에 의한 전력 사용량이 크게 증가하는 것을 관찰할 수 있으며, 다른 오버헤드 보다 큰 값을 가짐을 알 수 있다. 이것은 송신측과 마찬가지로 3DES의 특성에 기인한다. 시스템 호출과 softirq에서 데이터 증가에 따른 전력 사용량 증가는 각각 데이터 복사와 체크섬 계산에 의한 것이다.

그림 7은 1024B 데이터에 대해서 네트워크 수신과 복호화를 수행할 때 시스템 전체 전력 사용량과 프로세서 코어에서의 전력 사용량을 비교하고 있다. 프로세서 코어에서 사용하는 전력량은 시스템 전체 전력 소비량의 10~22%를 보이고 있다. 이러한 성향은 송신측과 크게 다르지 않음을 알 수 있다.

그림 8은 수신측에서 기능별 오버헤드를 보여주고 있다. 송신측과 같이 수신측에서도 전력 사용량 그래프와 오버헤드 그래프가 같은 성향을 보인다. 시스템 호출과 sofirq에서 데이터 크기에 따른 오버헤드의 증가가 명확히 보이지 않는 이유는 분석 결과 이들이 수행되는 중에 송신측으로부터 전송된 TCP의 FIN 메시지가 수신되었기 때문이다. 따라서 그림 8의 시스템 호출과 softirq는 FIN과 같은 TCP 제어 메시지를 위한 인터럽트 핸들링 및 softirq 오버헤드가 더해져 있으며, 그로 인해 측정 오차가 각각 존재한다. 그 예로서 그림 9는 시간에 따른 프로토콜 스택의 수행 중 일부를 보여주고 있다. 그림에서 볼 수 있는 바와 같이 softirq가 수행하고 있는 도중에 디바이스 드라이버의 인터럽트 핸들러가 238us 동안 수행하는 것을 알 수 있다. 이것은 TCP의 FIN 메시지가 도착하여 발생한 인터럽트를 처리하는 것이다. 그리고 그 이후에 softirq가 계속 수행되는 것을 볼 수 있는데, 이것이 앞서 도착한 데이터의 잔여 처리를 포함하는지 FIN 메시지를 위한 softirq 작업만을 포함하는지 현재는 구분이 불가능하다. 이러한 부분은 향후 더욱 자세한 측정을 통해서 구분해낼 수 있을 것이다.

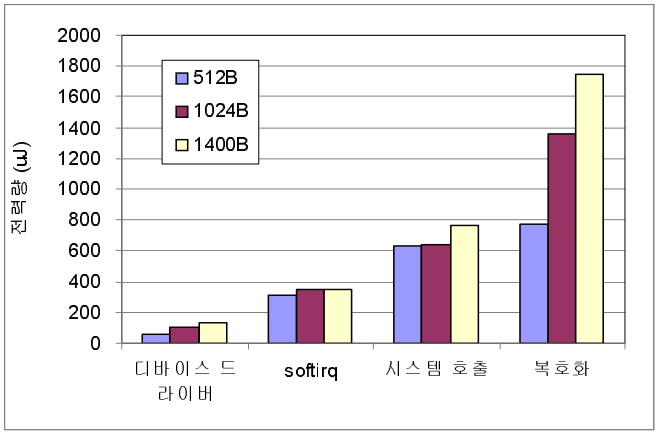

그림 6 수신측 작업별 프로세서 코어 전력 사용량

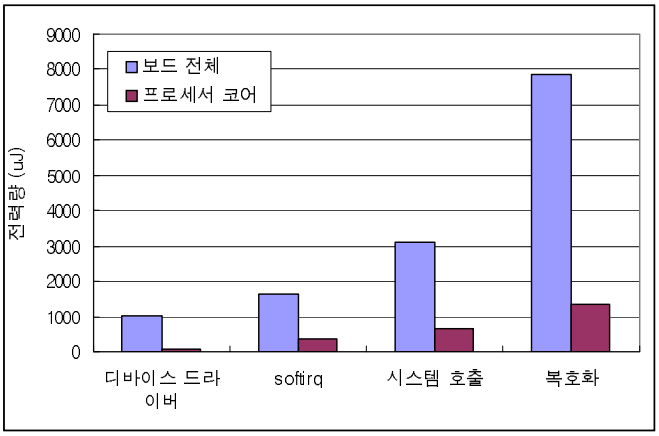

그림 7 수신측 시스템 전체 전력 사용량과 프로세서 코어 전력 사용량 비교 (1024B)

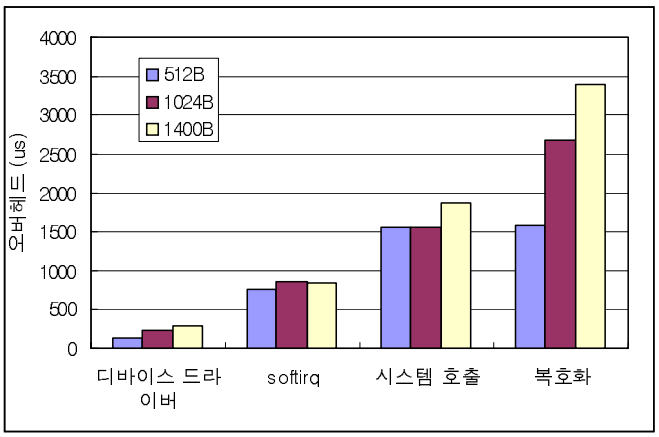

그림 8 수신측 작업별 수행 오버헤드

5. 결론 및 향후 연구 계획

본 논문은 임베디드 시스템에서 보안 데이터 송수신 시 발생하는 전력 사용량과 오버헤드를 세부 기능별로 구분하고 측정하였다. 측정 결과 3DES 기반의 암호화 및 복호화 작업 동안의 전력 사용량이 TCP/IP 통신을 위한 전력 사용량 보다 데이터 크기가 증가함에 따라 더욱 크게 측정되었다. 이것은 3DES 알고리즘이 DES 알고리즘을 3번 중복 수행하기 때문에 데이터 크기에 대해서 전력 사용량과 수행 오버헤드 모두 크게 영향 받는 것으로 분석되었다. 해당 작업의 수행 오버헤드 역시 유사한 성향을 보였다. 프로세서 코어의 전력 사용량은 시스템 전체 사용량의 10~22% 정도를 차지하는 것으로 측정되었다. 이러한 연구 결과는 임베디드 시스템에서 저전력 보안 데이터 전송을 위해 고려할 사항들을 정량적으로 제시하여 해당 연구 분야에 기여할 수 있을 것으로 기대된다.

향후 연구 계획으로는 프로세서 코어뿐만 아니라 다른 부분의 전력 소비량을 추가적으로 측정하고 무선 랜 환경에서 전력 측정을 수행하려고 한다. 또한 저전력 보안 데이터 전송이 가능한 기법을 연구 제안하려고 한다.

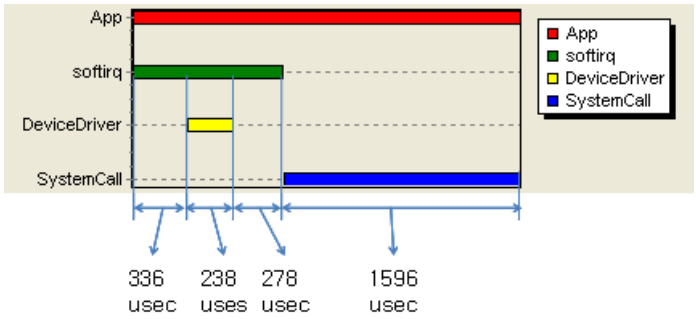

그림 9 수신측에서 시간 흐름에 따른 계층별 작업 수행

참고 문헌

[1] Alan O. Freier, Philip Karlton, and Paul C. Kocher, "The SSL Protocol (Version 3.0)," Transport Layer Security Working Group, INTERNET-DRAFT, 1996 November.

[2] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," Network Working Group, Request for Comments: 4301, December 2005.

[3] Harkirat Singh and Suresh Singh, "Energy Consumption of TCP Reno, Newreno, and SACK in Multi-Hop Wireless Networks," In Proceedings of ACM SIGMETRICS 2002, June 2002.

[4] Bokyung Wang and Suresh Singh, "Computational Energy Cost of TCP," In Proceedings of IEEE Infocom 2004, March 2004.

[5] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "Analyzing the Energy Csumption of Security Protocols," In Proceedings of ISLPED 2003, August 2002.

[6] Alireza Hodjat and Ingrid Verbauwhede, "The Energy Cost of Secrets in Ad-hoc Networks," In Proceedings of IEEE CAS Workshop on Wireless Communications and Networking, September 2002.

[7] Markus Jakobsson and David Pointcheval, "Mutual

Authentication for Low-Power Mobile Devices," Lecture Notes In Computer Science, Vol. 2339, pp. 178-195, 2002.
[8] D. S. Wong and A. H. Chan. "Mutual authentication and key exchange. for low power wireless communications," In Proceedings of IEEE MILCOM 2001, pp. 39-43, October 2001.
[9] Ramesh Karri and Piyush Mishra, "Minimizing Energy Consumption Of Secure Wireless Session With Qos Constraints," In Proceedings of ICC 2002, pp.2053-2057, May 2002.
[10] D. Clark, V. Jacobson, J. Romkey, and H. Salwen, "An Analysis of TCP Processing Overhead," IEEE Communications Magazine, 27(6), June 1989.
[11] Jonathan Kay and Joseph Pasquale, "Profiling and reducing processing overheads in TCP/IP," IEEE/ACM Transactions on Networking, Vol. 4, No. 6, pp. 817 - 828, December 1996.
[12] G. Xylomenos, G. C. Polyzos, P. Mahonen, M. Saaranen, "TCP performance issues over wireless links," IEEE Communications Magazine, Vol. 39, No. 4, pp. 52-58, April 2001.
[13] Hyun-Wook Jin and Chuck Yoo, "Impact of Protocol Overheads on Network Throughput over High-Speed Interconnects: Measurement, Analysis, and Improvement," The Journal of Supercomputing, Vol. 41, No. 1, pp. 17-40, July 2007.
[14] (주)휴인스, http://www.huins.com.
[15] National Instrument Corporation, http://www.ni.com.