(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0069606
(43) 공개일자 2011년06월23일

(51) Int. Cl.
G06F 9/28 (2006.01) *G06F 9/38* (2006.01)
G06F 9/50 (2006.01)
(21) 출원번호 10-2009-0126407
(22) 출원일자 2009년12월17일
심사청구일자 2009년12월17일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가 1
(72) 발명자
정용화
대전광역시 유성구 지족동 877 열매마을아파트 505-402
이성주
서울특별시 마포구 성산동 47-14
(뒷면에 계속)
(74) 대리인
나승택, 조영현

전체 청구항 수 : 총 8 항

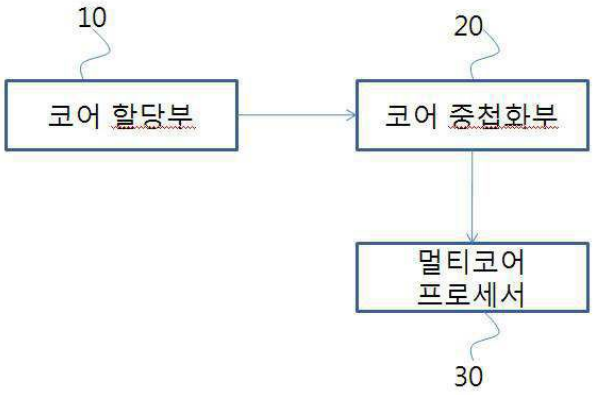
(54) 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 그 방법

(57) 요 약

본 발명은 멀티코어 프로세서를 이용하여 AES-CCM 데이터를 계산하고 이를 통신 처리하는 장치 및 방법에 관한 것으로, 특히 사용자 수준에서의 쓰레드-수준 병렬성을 이용하여 계산 과정과 전송 과정을 중첩하여 동시 처리하고, 동기화 오버헤드를 고려한 최적의 파이프라인 단계 수를 도출한 후, 계산 및 통신 처리를 중첩수행함으로써, 전체 수행시간을 감소시키는데 적당하도록 한 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 그 방법에 관한 것이다.

본 발명인 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치를 이루는 구성수단은, 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치에 있어서, 상기 멀티코어 프로세서의 코어들 중, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당하는 코어 할당부, 상기 코어 할당부에 의하여 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시키고, 상기 파이프라인의 각 단계 사이에 동기화를 삽입시키는 코어 중첩화부, 상기 코어 중첩화부의 중첩화에 따라 계산 과정과 통신 과정을 병렬 처리하는 멀티코어 프로세서를 포함하여 구성되는 것을 특징으로 한다.

대 표 도 - 도3



(72) 발명자

민병기

대전광역시 유성구 전민동 청구나래아파트 109동 606호

이은지

경상북도 경산시 정평동 130 현대타운 106-602

이 발명을 지원한 국가연구개발사업

과제고유번호 20090302185256

부처명 한국산업기술재단

연구관리전문기관

연구사업명 지역혁신인력양성사업

연구과제명 (1차년도)베터리로 동작하는 감시카메라용 비디오 압축 특성을 이용한 정보보호 및 상황 인지 연구를 통한 최적화

기여율

주관기관 고려대학교

연구기간 2009년 05월 01일 ~ 2010년 04월 30일

특허청구의 범위

청구항 1

멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치에 있어서,

상기 멀티코어 프로세서의 코어들 중, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당하는 코어 할당부;

상기 코어 할당부에 의하여 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시키고, 상기 파이프라인의 각 단계 사이에 동기화를 삽입시키는 코어 중첩화부;

상기 코어 중첩화부의 중첩화에 따라 계산 과정과 통신 과정을 병렬 처리하는 멀티코어 프로세서를 포함하여 구성되는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치.

청구항 2

청구항 1에 있어서,

상기 파이프라인의 총 단계 수는 각 단계 사이에 삽입되는 동기화 오버헤드를 고려하여 상기 계산 및 통신 처리에 걸리는 전체 수행시간을 최소화시키는 단계 수인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방장치.

청구항 3

청구항 2에 있어서,

상기 파이프라인 총 단계 수(m_{opt})는 $m_{opt} = \sqrt{\frac{MR_{\max}}{T_{syn}}}$ 에 의하여 결정되는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치.

여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, R_{max}는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, T_{syn}는 동기화 오버헤드임.

청구항 4

청구항 3에 있어서,

상기 결정된 파이프라인 총 단계 수가 소수인 경우, 상기 파이프라인 총 단계 수는 반올림된 자연수인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치.

청구항 5

청구항 1에 있어서,

상기 코어 중첩화부는 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread(POSIX Thread)를 이용하여 상기 계산 처리와 통신 처리를 중첩화시키는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치.

청구항 6

멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법에 있어서,

상기 멀티코어 프로세서의 코어들 중, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당하는 코어 할당 단계;

상기 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시키고, 상기 파이프라인의 각 단계 사이에 동기화를 삽입시키는 코어 중첩화 단계;

상기 중첩화에 따라 상기 멀티코어 프로세서가 상기 AES-CCM에 대한 계산 과정과 통신 과정을 병렬 처리하는 단계를 포함하여 구성되는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법.

청구항 7

청구항 6에 있어서,

상기 파이프라인 총 단계 수(m_{opt})는 $m_{opt} = \sqrt{\frac{MR_{\max}}{T_{syn}}}$ 에 의하여 결정되는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법.

여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, Rmax는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, Tsyn는 동기화 시간임.

청구항 8

청구항 7에 있어서,

상기 결정된 파이프라인 총 단계 수가 소수인 경우, 상기 파이프라인 총 단계 수는 반올림된 자연수인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 멀티코어 프로세서를 이용하여 AES-CCM 데이터를 계산하고 이를 통신 처리하는 장치 및 방법에 관한 것으로, 특히 사용자 수준에서의 쓰레드-수준 병렬성을 이용하여 계산 과정과 전송 과정을 중첩하여 동시 처리하고, 동기화 오버헤드를 고려한 최적의 파이프라인 단계 수를 도출한 후, 계산 및 통신 처리를 중첩수행함으로써, 전체 수행시간을 감소시키는데 적당하도록 한 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 그 방법에 관한 것이다.

배 경 기 술

[0002] 최근 MPEG 비디오 스트림 등 대용량 데이터의 이용이 증가함에 따라, 이러한 대용량 데이터 전송시 정보 보호가 중요한 문제로 부각되고 있다. 전송되는 데이터의 정보 보호는 기밀성의 보장과 더불어 HMAC(Keyed-Hash Message Authentication Code) 등과 같은 메시지 인증 기법을 추가적으로 적용하는 암호화 알고리즘을 활용하여 무결성도 동시에 보장할 필요가 있다.

[0003] 이러한 암호화 알고리즘은 높은 계산 요구량을 갖는다. 따라서 이들을 실시간으로 처리하기 위하여 전용 하드웨어 칩을 사용하는 것이 일반적인 추세이다.

[0004] 현재 가장 널리 사용되는 블록 암/복호화 표준은 AES(Advanced Encryption Standard)인데, AES의 다양한 운용 모드 중에서도 CCM(Counter with CBC-MAC)은 한 번의 수행만으로 기밀성과 무결성을 동시에 보장할 수 있는 장점 등으로 인하여 최근 많은 관심을 받는 방법이다. 이러한 AES-CCM을 멀티코어 프로세서를 활용하여 병렬화 한

다면, 전용 하드웨어 칩을 사용하는 현재 솔루션과 비교하여 저렴하면서 실용적으로 데이터의 기밀성과 무결성을 동시에 보장하는 솔루션을 확보할 수 있을 것으로 기대된다.

[0005] 도 1은 AES-CCM의 계산 특성을 보여주고 있다. CCM은 운용모드 중 하나인 CTR(Counter)을 이용하여 데이터의 기밀성을 제공하면서, CBC-MAC을 이용하여 메시지 인증 및 무결성도 동시에 제공한다. 먼저, CBC(Cipher Block Chaining)를 이용하여 MAC으로 사용할 암호화 블록을 만들고, CTR을 이용해 메시지를 암호화한다.

[0006] AES-CCM은 알려진 두 가지 표준 기술을 사용하기 때문에 신뢰성이 높다. 또한, CBC-MAC에서 인증과 암호화에 다른 키를 사용하는 것과는 달리, 한 개의 키로 두 가지 모두 사용 가능한 장점이 있다. CCM을 이용할 경우, 암호화를 할 필요 없이 무결성 검증만 필요로 하는 데이터가 포함된 경우 추가적인 암호문 오버헤드 없이 암호화할 수 있어 데이터를 다루기가 쉽다. 뿐만 아니라 CCM은 암호화하는 함수 하나만을 사용하여 암호화와 복호화를 모두 수행하기 때문에 비교적 작은 크기의 코드로 구현 할 수 있다. 이러한 특성의 CCM은 다른 운용모드들에 비해 융통성 있는 적용이 가능하게 해 준다.

[0007] 도 2는 멀티코어 환경에서 효과적인 병렬화를 위한 AES-CCM의 구조를 보여준다. Key 초기화(S210)와 메시지의 헤더 검증(S220)을 위한 수행은 작은 데이터를 처리하기 때문에 전체 수행시간에 큰 영향을 미치지 않는다. MAC 계산(S240)도 암호화 처리가 된 블록에 대하여 XOR 연산만 실행하기 때문에 병렬처리가 필수적이지는 않다. 그러나 메시지 암호화/인증 부분(S230)은 전체 메시지를 암호화/인증 블록의 개수만큼 처리해야 하기 때문에 전체 수행시간의 대부분을 차지한다. 따라서 이 부분에 대한 병렬화가 필요하다.

[0008] 메시지 암호화/인증은 CTR을 이용하여 메시지를 암호화하는 계산과 CBC-MAC을 이용하여 MAC 블록을 생성하는 계산을 수행한다. 이 두 과정은 모두 AES 암호화를 이용하고 같은 크기의 메시지를 처리하기 때문에 거의 비슷한 수행 시간이 소요되며, 이 두 부분을 효과적으로 병렬처리 해야 한다.

[0009] 전체 수행시간의 대부분을 차지하는 메시지 암호화/인증에서 데이터 수준 병렬화(Data-level parallelism)을 활용하기 위해서는 데이터 사이의 종속성이 없어야 한다. 그러나 CTR 모드와 달리, CBC-MAC의 경우 블록간의 체이닝으로 종속성이 존재하여 데이터 병렬화가 가능한 CTR만을 병렬처리 하는 방법을 생각할 수 있다.

[0010] 그러나 CTR의 메시지 암호화와 CBC-MAC 블록 생성의 수행시간이 거의 비슷하기 때문에 이러한 병렬화로는 수행시간을 크게 줄이지 못한다. 즉, 4-코어를 사용하는 경우 CTR은 1/4로 줄지만 CBC-MAC 블록 생성 시간은 감소하지 않기 때문에, 메시지 암호화/인증 부분의 speed-up은 대략 1.6에 불과하다.

[0011] 위의 문제를 해결하기 위해 CTR와 CBC-MAC을 함께 병렬처리, 즉 태스크 수준 병렬화(Task-Level parallelism)을 활용하는 방법을 생각할 수 있다. 즉, 병렬처리 구간에서 한 개의 코어는 CBC-MAC을 수행하고 나머지 코어들은 CTR를 수행하는 방법으로 병렬처리를 할 수 있다. 그러나 이러한 방법으로는 4-코어를 사용했을 때 speed-up이 최대 2 에 머무는 한계를 보인다. 더 좋은 성능을 얻기 위해서는 일반적인 병렬처리 기법 외에 추가적인 기법의 개발이 필요한 실정이다.

[0012] 즉, 멀티코어 프로세서의 활용 효율을 증대시키면서, AES-CCM의 처리 속도를 감소시킬 수 있는 새로운 병렬 기법이 필요한 실정이다.

발명의 내용

해결 하고자하는 과제

[0013] 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위하여 창안된 것으로, 사용자 수준에서의 쓰레드-수준 병렬성을 이용하여 계산 과정과 전송 과정을 중첩하여 동시 처리하고, 동기화 오버헤드를 고려한 최적의 파이프라인 단계 수를 도출한 후, 계산 및 통신 처리를 중첩수행함으로써, 전체 수행시간을 감소시킬 수 있는 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 그 방법을 제공하는 것을 그 목적으로 한다.

과제 해결수단

[0014] 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치를 이루는 구성수단은, 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치에 있어서, 상기 멀티

코어 프로세서의 코어들 중, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당하는 코어 할당부, 상기 코어 할당부에 의하여 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시키고, 상기 파이프라인의 각 단계 사이에 동기화를 삽입시키는 코어 중첩화부, 상기 코어 중첩화부의 중첩화에 따라 계산 과정과 통신 과정을 병렬 처리하는 멀티코어 프로세서를 포함하여 구성되는 것을 특징으로 한다.

[0015] 또한, 상기 파이프라인의 총 단계 수는 각 단계 사이에 삽입되는 동기화 오버헤드를 고려하여 상기 계산 및 통신 처리에 걸리는 전체 수행시간을 최소화시키는 단계 수인 것을 특징으로 한다.

[0016] 또한, 상기 파이프라인 총 단계 수(m_{opt})는 $$m_{opt} = \sqrt{\frac{MR_{\max}}{T_{syn}}}$$ 에 의하여 결정되는 것을 특징으로 한다. 여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, R_{max}는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, T_{syn}는 동기화 오버헤드이다.

[0017] 여기서, 상기 결정된 파이프라인 총 단계 수가 소수인 경우, 상기 파이프라인 총 단계 수는 반올림된 자연수인 것을 특징으로 한다.

[0018] 여기서, 상기 코어 중첩화부는 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread(POSIX Thread)를 이용하여 상기 계산 처리와 통신 처리를 중첩화시키는 것을 특징으로 한다.

[0019] 한편, 또 다른 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법을 이루는 구성수단은, 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 방법에 있어서, 상기 멀티코어 프로세서의 코어들 중, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당하는 코어 할당 단계, 상기 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시키고, 상기 파이프라인의 각 단계 사이에 동기화를 삽입시키는 코어 중첩화 단계, 상기 중첩화에 따라 상기 멀티코어 프로세서가 상기 AES-CCM에 대한 계산 과정과 통신 과정을 병렬 처리하는 단계를 포함하여 구성되는 것을 특징으로 한다.

[0020] 여기서, 상기 파이프라인 총 단계 수(m_{opt})는 $$m_{opt} = \sqrt{\frac{MR_{\max}}{T_{syn}}}$$ 에 의하여 결정되는 것을 특징으로 하고, 여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, Rmax는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, Tsyn는 동기화 시간이다.

[0021] 여기서, 상기 결정된 파이프라인 총 단계 수가 소수인 경우, 상기 파이프라인 총 단계 수는 반올림된 자연수인 것을 특징으로 한다.

효 과

[0022] 상기와 같은 과제 및 해결 수단을 가지는 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 방법에 의하면, 사용자 수준에서의 쓰레드-수준 병렬성을 이용하여 계산 과정과 전송 과정을 중첩하여 동시 처리하고, 동기화 오버헤드를 고려한 최적의 파이프라인 단계 수를 도출한 후, 계산 및 통신 처리를 중첩수행하기 때문에, 전체 수행시간을 감소시킬 수 있는 장점이 있다.

[0023] 또한, 데이터 수준에서 병렬처리 대신에, 쓰레드 수준에서 병렬처리를 수행함으로써, 멀티코어 프로세서의 사용 효율을 증대시킬 수 있는 장점이 있다.

발명의 실시를 위한 구체적인 내용

[0024] 이하, 첨부된 도면을 참조하여 상기와 같은 과제, 해결수단 및 효과를 가지는 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치 및 그 방법에 관한 바람직한 실시예를 상세하게 설명한다.

[0025] 도 3은 본 발명의 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치의 전체 구성도이

다.

[0026] 도 3에 도시된 바와 같이, 본 발명의 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 계산 및 처리 장치는 코어 할당부(10), 코어 중첩화부(20) 및 멀티코어 프로세서(30)를 포함하여 구성된다.

[0027] 상기 코어 할당부(10)는 상기 멀티코어 프로세서의 코어들 중에, 상기 AES-CCM 계산 및 통신을 위한 코어를 할당한다. 즉, 복수개의 코어 들 중에, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당한다.

[0028] 상기 AES-CCM은 종속성이 있는 데이터들을 포함하기 때문에, 멀티코어 프로세서의 코어들 중, 하나의 특정 코어를 통하여 계산 과정을 모두 수행하도록 코어를 할당할 수도 있고, 복수개의 코어를 통하여 계산 과정을 병렬 수행하도록 코어를 할당할 수도 있다. 이 계산 과정을 병렬 수행하는 경우에는 다양한 방법에 의하여 수행될 수 있고, 이 계산 과정을 병렬 수행하는 문제는 본 발명의 핵심 사항이 아니다. 즉, 본 발명의 핵심 사항은 종속성이 내재한 데이터를 계산하는 과정과 , 이 계산된 결과를 전송하는 과정을 중첩하여 병렬 처리하는 문제이다.

[0029] 상기와 같이 코어 할당부(10)에 의하여 AES-CCM 계산 처리를 위한 계산 코어와 이 계산된 결과를 전송하기 위한 통신 코어가 할당되면, 상기 코어 중첩화부(20)는 상기 코어 할당부(10)에 의하여 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시킨다.

[0030] 또한, 상기 코어 중첩화부(20)는 서로 다른 코어에서 상기 AES-CCM 계산 처리와 통신 처리가 중첩되어 동시 처리될 수 있도록, 상기 파이프라인의 각 단계 사이에 동기화를 삽입하는 기능을 수행한다. 즉, 상기 파이프라인 각 단계를 통하여 계산 또는 통신 절차가 수행된 후, 다음 단계를 통하여 계산 및 통신 절차가 동시 수행될 수 있도록, 상기 코어 중첩화부(20)는 파이프라인 각 단계 사이에 동기화를 삽입한다.

[0031] 상기 코어 중첩화부(20)는 상기 계산될 AES-CCM의 전체 데이터를 복수개의 파이프라인 단계와 동일하게 나누는 동작을 수행하고, 각 파이프라인의 단계에서 해당 코어를 통하여 계산될 수 있도록 제어한다.

[0032] 상기 코어 중첩화부(20)를 통하여 계산 과정과 통신 과정이 중첩화되면, 상기 멀티코어 프로세서(30)는 상기 코어 중첩화부의 중첩화에 따라 계산 과정과 통신 과정을 병렬 처리한다. 이 때, 계산 코어에 의하여 파이프라인 특정 단계에서 계산이 완료되면, 상기 계산 코어에서 다음 파이프라인 단계를 통하여 다음 계산을 수행하고, 상기 계산 코어에 의하여 파이프라인 특정 단계에서 계산된 결과를 상기 통신 코어를 통하여 전송하기 위하여, 상기 파이프라인 특정 단계와 다음 단계 사이에 동기화가 삽입될 수 있도록 상기 코어 중첩화부(20)가 제어한다.

[0033] 본 발명에서는 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread[B. Barney, POSIX Threads Programming, http://www.llnl.gov/computing/tutorials/pthreads, 2006.]를 이용하여 계산과 통신을 중첩시키고, 이를 응용 프로그램 수준의 파이프라인(pipeline) 설계 문제로 해석한다.

[0034] 특히, 파이프라인 단계(stage) 수를 증가시키면 비례적으로 성능이 개선되는 일반적인 문제와 달리, 계산 과정을 담당하는 코어와 통신 과정을 담당하는 코어를 매 단계마다 동기화(barrier synchronization) 시키는 오버헤드를 감안해야 한다.

[0035] 2-코어 시스템에서 각각의 코어에 계산 과정과 통신 과정을 할당하여 중첩 수행시키는 모습을 정리하면 도 4와 같다. 이때 한 번의 계산 단계가 수행되어야 해당되는 전송 단계가 수행될 수 있기 때문에, 도 4와 같이 각 단계마다 동기화가 추가된 파이프라인(Pipeline) 형태로 계산-통신 과정을 중첩하여 병렬처리가 가능하다.

[0036] 일반적인 병렬처리에서 공유 데이터를 동시에 접근하려고 할 때 발생하는 문제를 해결하기 위하여 동기화를 적용해야 한다. 통상 프로세서들이 하나의 칩 내에 구현된 멀티코어에서는 동기화 오버헤드가 적을 것으로 예상하나, 실제로 측정한 결과는 상당한 오버헤드가 있다. 따라서 전체 수행시간을 줄이기 위해 파이프라인 단계 수를 늘려야 하는 요구와 단계간 동기화 오버헤드를 줄이기 위해서는 단계 수를 줄여야 하는 상충 관계에서 최적의 단계 수를 구해야 하는 문제가 발생한다.

[0037] 즉, 상기 파이프라인의 최적의 총 단계 수는 각 단계 사이에 삽입되는 동기화 오버헤드를 고려하여, 상기 계산 및 통신 처리에 걸리는 전체 수행시간을 최소화시키는 단계 수가 될 수 있도록 최적화될 필요가 있다.

[0038] 도 4에서 보여주는 2-코어 시스템에서 계산 처리와 통신 처리를 중첩한 경우, 전체 데이터의 크기가 M(단위: 바이트)이고 계산-통신 과정 각각을 m번의 단계로 구분하여 중첩하는 경우, 다음과 같은 수식 1로 전체 수행 시간을 나타낼 수 있다.

[0039] $$(m+(n-1))\frac{M}{m}R_{\max}+mT_{syn}$$ (수식 1)

[0040] 여기서, n은 AES-CCM의 계산 및 통신 처리를 위하여 할당된 총 코어 수이고, R_{max}는 계산 또는 통신 과정 중 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표(단위 초/바이트)이며, T_{syn}은 동기화 오버헤드 시간이다.

[0041] 상기 수식 1은 AES-CCM에 대한 계산 및 통신 처리를 수행하는 총 코어 수가 n인 경우의 수식이다. 즉, 도 4에서 보여준 바와 같이, 총 2 개의 코어를 이용하여, 하나의 코어에는 계산 과정을 수행하고, 다른 하나의 코어에서는 통신 과정을 수행하는 시스템에서의 전체 수행 시간은 아래 수식 2와 같다.

[0042] $$(m+1)\frac{M}{m}R_{\max}+mT_{syn}$$ (수식 2)

[0043] 여기서 R_{max}는 계산 또는 통신 과정 중 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표(단위: 초/바이트)이다. 즉, AES-CCM으로 데이터를 암호화하는데 소요되는 처리 능력을 R_{comp}, 암호화된 데이터를 통신 선로로 전송하는데 소요되는 처리 능력을 R_{comm}으로 표시할 때, R_{max} = max(R_{comp}, R_{comm})이 된다.

[0044] 그러면, Rmax에 전체 데이터 M을 m등분한 만큼 곱하면 하나의 단계를 수행하는데 걸리는 시간을 얻을 수 있다. 따라서 전체 단계를 수행하는데 걸리는 시간은 총 단계 수인 (m+1) 만큼을 곱해주어야 한다. 또한, 전체 단계 수행시간에 동기화 오버헤드 T_{syn}를 m번 더해주어야 최종적으로 전체 수행 시간이 된다.

[0045] 최종적으로 상기 수식 1 또는 수식 2에 대한 최소화 문제를 풀면 최적 m_{opt}는 다음과 같다. 즉, 전체 수행 시간을 최소화하는 파이프라인 총 단계 수(m_{opt})를 구하면 다음 수식 3과 같다.

[0046] $$m_{opt}=\sqrt{\frac{MR_{\max}}{T_{syn}}}$$ (수식 3)

[0047] 즉, AES-CCM에 대한 계산 및 통신을 처리하기 위한 총 코어 수와 무관하게 총 처리 시간을 최소화하는 파이프라인 총 단계 수는 상기 수식 3과 같다.

[0048] 여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, R_{max}는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, T_{syn}는 동기화 오버헤드이다.

[0049] 한편, 상기 수식 3에 의하여 계산된 결과는 자연수가 아닌 소수일 수도 있다. 따라서, 이 경우, 상기 수식 3에 의하여 계산되어 결정된 파이프라인 총 단계 수가 소수인 경우에는, 상기 파이프라인 총 단계 수는 계산된 소수가 반올림된 자연수이다.

[0050] 한편, 본 발명에서의 코어 중첩화부는 멀티코어 프로세서를 통하여 계산 과정과, 계산된 결과를 전송하는 통신 과정을 중첩시켜 처리하기 위하여, 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread(POSIX Thread)를 이용하여 상기 계산 처리와 통신 처리를 중첩화시킨다.

[0051] 다음은, 상기와 같은 구성으로 이루어진 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치를 통한 통신 처리 방법에 대하여 상세하게 설명한다.

[0052] 먼저, 멀티코어 프로세서를 가지는 시스템에서, 상기 멀티코어 프로세서의 코어들 중에, 상기 AES-CCM 계산 및 통신을 위한 코어를 할당하는 코어 할당 단계가 수행된다. 즉, 복수개의 코어 들 중에, 상기 AES-CCM 계산을 처리하기 위한 계산 코어를 할당하고, 상기 계산 코어에 의하여 계산된 결과를 전송하기 위한 통신 코어를 할당한다.

[0053] 상기 AES-CCM은 종속성이 있는 데이터들을 포함하기 때문에, 멀티코어 프로세서의 코어들 중, 하나의 특정 코어를 통하여 계산 과정을 모두 수행하도록 코어를 할당할 수도 있고, 복수개의 코어를 통하여 계산 과정을 병렬 수행하도록 코어를 할당할 수도 있다. 이 계산 과정을 병렬 수행하는 경우에는 다양한 방법에 의하여 수행될 수

있고, 이 계산 과정을 병렬 수행하는 문제는 본 발명의 핵심 사항이 아니다. 즉, 본 발명의 핵심 사항은 종속성이 내재한 데이터를 계산하는 과정과 , 이 계산된 결과를 전송하는 과정을 중첩하여 병렬 처리하는 문제이다.

[0054] 상기와 같이 코어 할당 단계에서 AES-CCM 계산 처리를 위한 계산 코어와 이 계산된 결과를 전송하기 위한 통신 코어가 할당되면, 코어 중첩화 단계가 수행된다. 상기 코어 중첩화 단계는 상기 코어 할당 단계에서 할당된 코어들에 의하여 AES-CCM에 대한 계산 처리와 통신 처리 과정이 파이프라인 구조로 병렬 처리되도록 중첩화시킨다.

[0055] 또한, 상기 코어 중첩화 단계에서는 서로 다른 코어에서 상기 AES-CCM 계산 처리와 통신 처리가 중첩되어 동시 처리될 수 있도록, 상기 파이프라인의 각 단계 사이에 동기화를 삽입하는 기능을 수행한다. 즉, 상기 파이프라인 각 단계를 통하여 계산 또는 통신 절차가 수행된 후, 다음 단계를 통하여 계산 및 통신 절차가 동시 수행될 수 있도록, 상기 코어 중첩화 단계에서는 파이프라인 각 단계 사이에 동기화를 삽입한다.

[0056] 상기 코어 중첩화 단계에서는 상기 계산될 AES-CCM의 전체 데이터를 복수개의 파이프라인 단계와 동일하게 나누는 동작을 수행하고, 각 파이프라인의 단계에서 해당 코어를 통하여 계산될 수 있도록 제어한다.

[0057] 상기 코어 중첩화 단계를 통하여 계산 과정과 통신 과정이 중첩화되면, 상기 멀티코어 프로세서는 상기 코어 중첩화부의 중첩화에 따라 계산 과정과 통신 과정을 병렬 처리한다. 이 때, 계산 코어에 의하여 파이프라인 특정 단계에서 계산이 완료되면, 상기 계산 코어에서 다음 파이프라인 단계를 통하여 다음 계산을 수행하고, 상기 계산 코어에 의하여 파이프라인 특정 단계에서 계산된 결과를 상기 통신 코어를 통하여 전송하기 위하여, 상기 파이프라인 특정 단계와 다음 단계 사이에 동기화가 삽입될 수 있도록 상기 코어 중첩화 단계에서 추가적으로 제어한다.

[0058] 본 발명에서는 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread[B. Barney, POSIX Threads Programming, http://www.llnl.gov/computing/tutorials/pthreads, 2006.]를 이용하여 계산과 통신을 중첩시키고, 이를 응용 프로그램 수준의 파이프라인(pipeline) 설계 문제로 해석한다.

[0059] 특히, 파이프라인 단계(stage) 수를 증가시키면 비례적으로 성능이 개선되는 일반적인 문제와 달리, 계산 과정을 담당하는 코어와 통신 과정을 담당하는 코어를 매 단계마다 동기화(barrier synchronization) 시키는 오버헤드를 감안해야 한다.

[0060] 2-코어 시스템에서 각각의 코어에 계산 과정과 통신 과정을 할당하여 중첩 수행시키는 모습을 정리하면 도 4와 같다. 이때 한 번의 계산 단계가 수행되어야 해당되는 전송 단계가 수행될 수 있기 때문에, 도 4와 같이 각 단계마다 동기화가 추가된 파이프라인(Pipeline) 형태로 계산-통신 과정을 중첩하여 병렬처리가 가능하다.

[0061] 일반적인 병렬처리에서 공유 데이터를 동시에 접근하려고 할 때 발생하는 문제를 해결하기 위하여 동기화를 적용해야 한다. 통상 프로세서들이 하나의 칩 내에 구현된 멀티코어에서는 동기화 오버헤드가 적을 것으로 예상하나, 실제로 측정한 결과는 상당한 오버헤드가 있다. 따라서 전체 수행시간을 줄이기 위해 파이프라인 단계 수를 늘려야 하는 요구와 단계간 동기화 오버헤드를 줄이기 위해서는 단계 수를 줄여야 하는 상충 관계에서 최적의 단계 수를 구해야 하는 문제가 발생한다.

[0062] 즉, 상기 파이프라인의 최적의 총 단계 수는 각 단계 사이에 삽입되는 동기화 오버헤드를 고려하여, 상기 계산 및 통신 처리에 걸리는 전체 수행시간을 최소화시키는 단계 수가 될 수 있도록 최적화될 필요가 있다.

[0063] 도 4에서 보여주는 2-코어 시스템에서 계산 처리와 통신 처리를 중첩한 경우, 전체 데이터의 크기가 M(단위: 바이트)이고 계산-통신 과정 각각을 m번의 단계로 구분하여 중첩하는 경우, 다음과 같은 수식 1로 전체 수행 시간을 나타낼 수 있다.

[0064] $$(m+(n-1))\frac{M}{m}R_{\max}+mT_{syn} \quad \text{(수식 1)}$$

[0065] 여기서, n은 AES-CCM의 계산 및 통신 처리를 위하여 할당된 총 코어 수이고, R_{max}는 계산 또는 통신 과정 중 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표(단위 초/바이트)이며, T_{syn}은 동기화 오버헤드 시간이다.

[0066] 상기 수식 1은 AES-CCM에 대한 계산 및 통신 처리를 수행하는 총 코어 수가 n인 경우의 수식이다. 즉, 도 4에서 보여준 바와 같이, 총 2 개의 코어를 이용하여, 하나의 코어에는 계산 과정을 수행하고, 다른 하나의 코어에서

는 통신 과정을 수행하는 시스템에서의 전체 수행 시간은 아래 수식 2와 같다.

[0067] $$(m+1)\frac{M}{m}R_{\max}+mT_{syn}$$ (수식 2)

[0068] 여기서 R_{max}는 계산 또는 통신 과정 중 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표(단위: 초/바이트)이다. 즉, AES-CCM으로 데이터를 암호화하는데 소요되는 처리 능력을 R_{comp}, 암호화된 데이터를 통신 선로로 전송하는데 소요되는 처리 능력을 R_{comm}으로 표시할 때, R_{max} = max(R_{comp}, R_{comm})이 된다.

[0069] 그러면, Rmax에 전체 데이터 M을 m등분한 만큼 곱하면 하나의 단계를 수행하는데 걸리는 시간을 얻을 수 있다. 따라서 전체 단계를 수행하는데 걸리는 시간은 총 단계 수인 (m+1) 만큼을 곱해주어야 한다. 또한, 전체 단계 수행시간에 동기화 오버헤드 T_{syn}를 m번 더해주어야 최종적으로 전체 수행 시간이 된다.

[0070] 최종적으로 상기 수식 1 또는 수식 2에 대한 최소화 문제를 풀면 최적 m_{opt}는 다음과 같다. 즉, 전체 수행 시간을 최소화하는 파이프라인 총 단계 수(m_{opt})를 구하면 다음 수식 3과 같다.

[0071] $$m_{opt}=\sqrt{\frac{MR_{\max}}{T_{syn}}}$$ (수식 3)

[0072] 즉, AES-CCM에 대한 계산 및 통신을 처리하기 위한 총 코어 수와 무관하게 총 처리 시간을 최소화하는 파이프라인 총 단계 수는 상기 수식 3과 같다.

[0073] 여기서, M은 계산 및 통신 처리될 전체 데이터의 크기이고, R_{max}는 계산 또는 통신 과정 중, 수행 시간이 더 오래 걸리는 과정의 처리 또는 전송 능력 지표이며, T_{syn}는 동기화 오버헤드이다.

[0074] 한편, 상기 수식 3에 의하여 계산된 결과는 자연수가 아닌 소수일 수도 있다. 따라서, 이 경우, 상기 수식 3에 의하여 계산되어 결정된 파이프라인 총 단계 수가 소수인 경우에는, 상기 파이프라인 총 단계 수는 계산된 소수가 반올림된 자연수이다.

[0075] 한편, 본 발명에서의 코어 중첩화부는 멀티코어 프로세서를 통하여 계산 과정과, 계산된 결과를 전송하는 통신 과정을 중첩시켜 처리하기 위하여, 사용자 수준에서의 쓰레드-수준 병렬성(Thread-Level Parallelism)을 제공하는 Pthread(POSIX Thread)를 이용하여 상기 계산 처리와 통신 처리를 중첩화시킨다.

[0076] 다음은 본 발명에서 제안된 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리를 중첩화시켜 수행한 효용성을 검증한다. 이를 위하여, Intel Core2 Quad CPU 2.33GHz의 실험환경에서 제안 방법의 효용성을 검증하였다. 먼저 코어 간 동기화 오버헤드 T_{syn}를 확인하기 위해 Pthread barrier 명령어를 이용하여 측정한 결과, 2-코어 경우 약 10.1㎲, 4-코어의 경우 약 49.2㎲의 상당한 오버헤드가 있음을 확인하였다. 다음으로 다양한 데이터 크기 별로 AES-CCM 수행 시간을 측정한 결과, 도 5와 같은 결과를 얻었다. 즉, R_{comp} = 0.037 (㎲/Byte)의 결과를 얻었다.

[0077] 본 실험에서는 멀티코어 CPU에 다양한 유무선 네트워크 표준이 접속되는 것으로 가정하고, 각 유무선 표준의 최대 전송 능력의 일부가 실질적인 전송 능력이 됨을 감안하여, 전송 능력을 다음과 같은 범위로 설정하였다. 즉, 10Mbps, 100Mbps, 1000Mbps에 해당하도록 R_{comm}을 0.8 ㎲/Byte, 0.08 ㎲/Byte, 0.008 ㎲/Byte로 설정하였다.

[0078] 데이터 크기 M이 10MB와 10GB의 경우에 대하여, 수식 3에 의한 최적 파이프라인 총 단계 수(m_{opt})와 해당되는 speed-up(= 순차 수행 시간/계산-통신 중첩에 의한 수행 시간)을 정리하면 다음 표 1과 같다. 여기서 순차 수행 시간은 MR_{comp} + MR_{comm}으로 계산하였다.

[0079]

	10Mbps (0.8 us/B)	100Mbps (0.08 us/B)	1000Mbps (0.008 us/B)
M=10MB	1.04 (m_{opt} = 894)	1.45 (m_{opt} = 292)	1.20 (m_{opt} = 192)

M=10GB	1.04 (m_{opt} = 28,284)	1.46 (m_{opt} = 8,944)	1.21 (m_{opt} = 6,082)

[0080] - 표 1 -

[0081] 상기 표 1의 결과에서, 네트워크 성능이 증가할수록 최적 m_{opt}는 적어지고, 처리 또는 전송해야 할 데이터의 크기가 증가할수록 최적 m_{opt}는 증가함을 알 수 있다. 특이한 점은 1000Mbps의 경우 이미 R_{comp}(0.037 ㎲/B)가 R_{comm}(0.008 ㎲/B) 보다 커져서 R_{max}가 R_{comp}이 되며, 이러한 현상은 네트워크 성능이 더욱 증가하여도 적용된다.

[0082] 또한, 100Mbps에서 가장 높은 speed-up을 달성할 수 있는데, 이는 10Mbps나 1000Mbps의 경우에 비하여 R_{comm}과 R_{comp}의 차이가 적기 때문이다. 만약 최적의 네트워크 환경(즉, R_{comm}이 R_{comp}와 같아져 각 단계 수행 시 계산 및 통신 과정의 대기 시간이 없고 완벽한 부하 분산을 얻는 환경)을 가정하면, 최대 speed-up = 1.98(즉, =10MB인 경우 m_{opt} = 192에서) 또는 2.00(즉, =10GB인 경우 m_{opt} = 6,082에서)를 기대할 수 있다.

[0083] 마지막으로, 본 발명 및 본 실험에서는 설명의 편의상 2-코어 시스템을 가정하였으나, 더 많은 코어에서도 효과적으로 성능개선을 기대할 수 있다. 즉, 계산 과정에서 더 많은 쓰레드-레벨 병렬성을 확보하기 위하여 일반적인 병렬처리 기법(앞서 종래 기술에서 설명한 병렬 처리 기법, 예를 들어, CTR 만을 병렬 처리하는 기법 또는 CTR과 CBC-MAC을 태스크 수준에서 병렬화하여 동시 병행 처리하는 기법 등)을 추가로 적용하거나 복수의 네트워크 패스를 적용하면 응용 프로그램 수준에서 multiple-issue 파이프라인[J. Hennessy and D. Patterson, *Computer* Architecture: A Quantitative Approach - Fourth Edition, Elsevier, 2007.]과 같은 병렬처리가 가능하다. 또한, 이러한 추가 병렬성 확보는 앞서 언급한 R_{comm}과 R_{comp}의 균형을 맞추기 위한 용도로도 사용될 수 있다.

도면의 간단한 설명

[0084] 도 1은 본 발명에 적용되는 AES-CCM의 처리 순서도이다.

[0085] 도 2는 본 발명에 적용되는 AES-CCM의 계산 특성을 설명하기 위한 이해도이다.

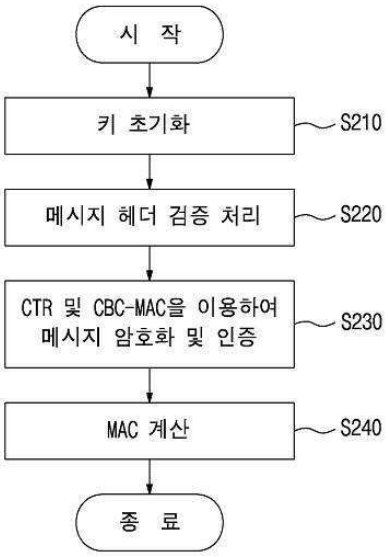
[0086] 도 3은 본 발명의 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 계산 및 통신 처리 장치의 구성도이다.

[0087] 도 4는 본 발명의 실시예에 따른 2-코어 시스템에서의 계산 및 통신 중첩화 병렬 처리를 설명하기 위한 예시도이다.

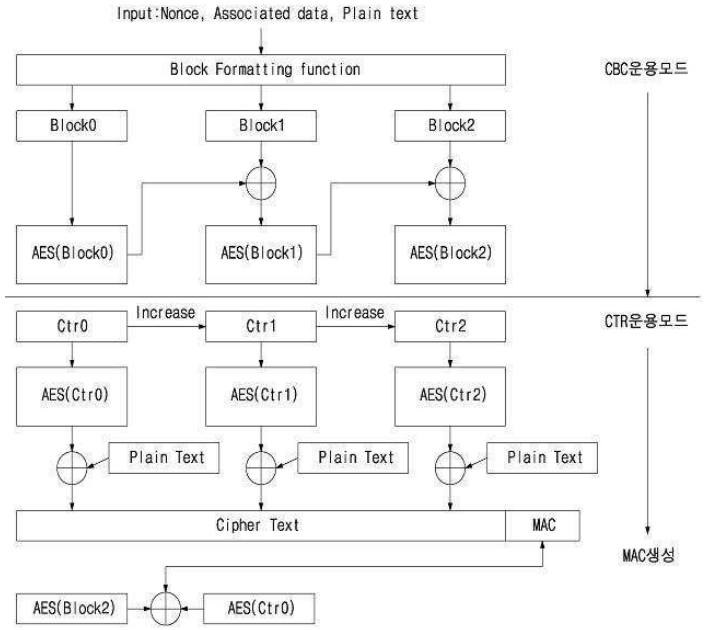
[0088] 도 5는 본 발명의 성능을 설명하기 위한 AES-CCM 처리 시간 그래프이다.

도면

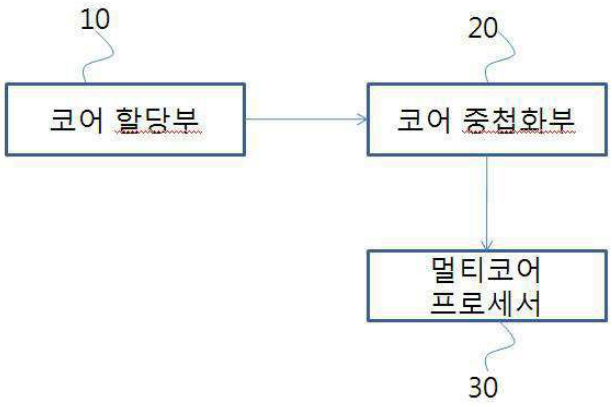
도면1


시 작
키 초기화
S210
메시지 헤더 검증 처리
S220
CTR 및 CBC-MAC을 이용하여
메시지 암호화 및 인증
S230
MAC 계산
S240
종 료

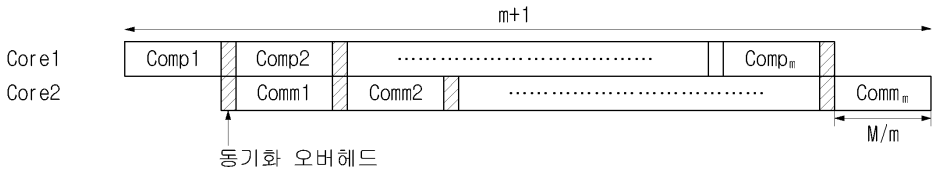
도면2


Input:Nonce, Associated data, Plain text
Block Formatting function
CBC운용모드
Block0
Block1
Block2
AES(Block0)
AES(Block1)
AES(Block2)
Ctr0
Increase
Ctr1
Increase
Ctr2
CTR운용모드
AES(Ctr0)
AES(Ctr1)
AES(Ctr2)
Plain Text
Plain Text
Plain Text
Cipher Text
MAC
MAC생성
AES(Block2)
AES(Ctr0)

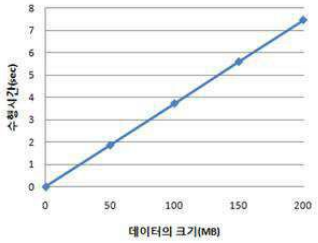
도면3


10
20
코어 할당부
코어 중첩화부
멀티코어
프로세서
30

도면4


m+1
Core1
Core2
Comp1
Comp2
Compm
Comm1
Comm2
Commm
M/m
동기화 오버헤드

도면5


수행시간(sec)
데이터의 크기(MB)
0
1
2
3
4
5
6
7
8
0
50
100
150
200