(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0070247
(43) 공개일자 2011년06월24일

(51) Int. Cl.
G06F 9/46 (2006.01) *G06F 9/38* (2006.01)
(21) 출원번호 10-2009-0126990
(22) 출원일자 2009년12월18일
심사청구일자 2009년12월18일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가 1
(72) 발명자
정용화
대전광역시 유성구 지족동 877 열매마을아파트 505-402
민병기
대전광역시 유성구 전민동 청구나래아파트 109동 606호
(뒷면에 계속)
(74) 대리인
나승택, 조영현

전체 청구항 수 : 총 14 항

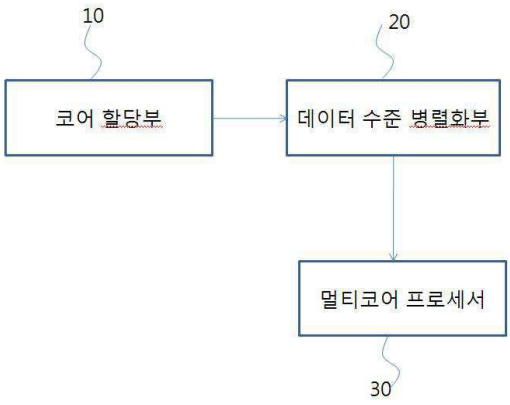
(54) 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법

(57) 요 약

본 발명은 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, 멀티코어 프로세서를 이용하여 AES-CCM 병렬처리 장치 및 그 방법에 관한 것으로, AES-CCM의 계산 특성과 멀티코어 프로세서의 동기화 오버헤드를 고려하여 데이터 수준 병렬화와 태스크 수준 병렬화를 동시에 활용함으로써, 수행시간을 효과적으로 감소시킬 수 있는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법에 관한 것이다.

본 발명인 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치를 이루는 구성수단은, AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당부, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어들에 의하여 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당하되, 상기 CTR 병렬 처리 및 CBC-MAC 병렬 처리를 순차적으로 수행시키는 데이터 수준 병렬화부, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 멀티코어 프로세서를 포함하여 이루어진 것을 특징으로 한다.

대 표 도 - 도5



(72) 발명자

이성주

서울특별시 마포구 성산동 47-14

이은지

경상북도 경산시 정평동 130 현대타운 106-602

이 발명을 지원한 국가연구개발사업

과제고유번호

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

AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당부;

상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어들에 의하여 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당하되, 상기 CTR 병렬 처리 및 CBC-MAC 병렬 처리를 순차적으로 수행시키는 데이터 수준 병렬화부;

상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 멀티코어 프로세서를 포함하여 이루어진 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 2

청구항 1에 있어서,

상기 코어 할당부는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당하고, 상기 CBC-MAC을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당하는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 3

청구항 2에 있어서,

상기 코어 할당부는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 4

AES-CCM의 처리 대상 태스크인 CTR를 병렬처리 하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당부;

상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어(들)에 의하여 순차 또는 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 할당하되, 상기 CTR 병렬 처리와 상기 CBC-MAC 순차 또는 병렬 처리를 병행적으로 수행시키는 데이터 및 태스크 수준 병렬화부;

상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 멀티코어 프로세서를 포함하여 이루어진 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 5

청구항 4에 있어서,

상기 코어 할당부는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당하고, 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 나머지 코어들에서 선택하여 할당하되, 순차 처리인 경우에는 1개의 코어를 할당하고 병렬 처리인 경우에는 2개 이상의 코어들을 할당하는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 6

청구항 5에 있어서,

상기 코어 할당부는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 7

청구항 2 또는 청구항 5에 있어서,

상기 코어 할당부는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치.

청구항 8

멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계;

상기 컴퓨터 시스템에서, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어들에 의하여 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당하되, 상기 CTR 병렬 처리 및 CBC-MAC 병렬 처리가 순차적으로 수행되도록 지시하는 데이터 수준 병렬화 단계;

상기 멀티코어 프로세서에서, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 순차적으로 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 단계를 포함하여 이루어진 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 9

청구항 8에 있어서,

상기 코어 할당 단계는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당하고, 상기 CBC-MAC을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당하는 단계인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 10

청구항 9에 있어서,

상기 코어 할당 단계는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 단계인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 11

멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, AES-CCM의 처리 대상 태스크인 CTR를 병렬처리 하고, CBC-

MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계;

상기 컴퓨터 시스템에서, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어(들)에 의하여 순차 또는 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 할당하되, 상기 CTR 병렬 처리와 상기 CBC-MAC 순차 또는 병렬 처리가 병행적으로 수행되도록 지시하는 데이터 및 태스크 수준 병렬화 단계;

상기 멀티코어 프로레서에서, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 단계를 포함하여 이루어진 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 12

청구항 11에 있어서,

상기 코어 할당 단계는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당하고, 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 나머지 코어들에서 선택하여 할당하되, 순차 처리인 경우에는 1개의 코어를 할당하고 병렬 처리인 경우에는 2개 이상의 코어들을 할당하는 단계인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 13

청구항 12에 있어서,

상기 코어 할당 단계는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 단계인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

청구항 14

청구항 9 또는 청구항 12에 있어서,

상기 코어 할당 단계는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 단계인 것을 특징으로 하는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, 멀티코어 프로세서를 이용하여 AES-CCM 병렬처리 장치 및 그 방법에 관한 것으로, AES-CCM의 계산 특성과 멀티코어 프로세서의 동기화 오버헤드를 고려하여 데이터 수준 병렬화와 태스크 수준 병렬화를 동시에 활용함으로써, 수행시간을 효과적으로 감소시킬 수 있는 멀티코어 프로세서를 이용하여 AES-CCM 병렬처리 장치 및 그 방법에 관한 것이다.

배 경 기 술

[0002] 디지털 정보사회가 고도화되고 전자상거래가 활성화됨에 따라, 암호/인증 기술은 현재, 인터넷을 기반으로 한 사회/경제적 활동의 안전성과 신뢰성, 그리고 사용자 프라이버시 보호 등을 위한 핵심 기술로서 인식되고 있다. 또한, 최근 MPEG 비디오 스트림 등 대용량 데이터 이용이 증가함에 따라, 이러한 암호/인증 기술을 이용한 대용량 데이터 정보 보호 문제가 이슈화되고 있다.

[0003] 대용량 데이터 정보 보호를 위해서는, 데이터의 기밀성(confidentiality)을 보장하는 기법은 물론, 그 외에 HMAC(Keyed-Hash Message Authentication Code) 등과 같은 메시지 인증 기법을 추가적으로 적용할 필요가 있다. 즉, 메시지 인증을 할 수 있는 암호화 알고리즘을 적용하여 대용량 데이터의 무결성(Integrity)을 동시에 보장해야 한다.

[0004] 이와 같이, 데이터의 기밀성 및 무결성을 동시에 보장하기 위한 방법으로는, 블록 암호화 운용모드(Block mode of operation)를 이용하는 방법이 있다.

[0005] 미국 국립표준기술원(NIST: National Institute of Standards and Technology)이 차세대 국제 표준 암호로서 채택한 AES(Advanced Encryption Standard)는 데이터의 무결성과 기밀성을 보장하기 위해 5가지 운용모드를 표준으로 규정하고 있다. 즉, ECB(Electronic Code Book), CBC(Cipher Block Chaining), CFB(Cipher FeedBack), OFD(Output FeedBack) 및 CTR(Counter)을 규정하고 있다.

[0006] 상기 다양한 표준 운영모드들 중 CTR 모드와 CBC-MAC 모드를 조합한 AES-CCM(Counter with CBC-MAC)은, 상기 조합된 두 모드의 기능이 결합되어 하나의 솔루션에서 암호화 및 메시지 무결성을 제공한다. 즉, 기밀성 기능과 키 관리(Key Management) 기능, 메시지 무결성(Message Integrity) 기능을 동시에 수행할 수 있다. 이러한 AES-CCM은, 무선랜(Wireless LAN) 보안 표준인 IEEE 802.11i와 WPA에서 공통으로 채택하고 있으며, WPA에서는 선택사항으로 규정되어 있고, WPA2에서는 강제 사항으로 규정되어 있다.

[0007] 한편, 상기 AES-CCM은, 위에서 언급한 바와 같이 이미 검증된 두 가지 표준 기술을 사용하기 때문에 높은 신뢰성을 나타내는 이점이 있다. 또한, 일반적으로 인증 및 암호화에 다른 키를 사용하는 것과는 달리, 하나의 키만을 사용할 수 있는 장점이 있다. 또한, 암호화 없이 무결성 검증만을 필요로 하는 데이터가 포함된 경우에도 AES-CCM을 이용하면, 추가적인 암호문 오버헤드 없이 암호화할 수 있어 데이터를 다루기가 용이하다는 장점이 있다. 나아가, AES-CCM은 암호화 함수 하나만을 사용하여 암호화 및 복호화를 모두 수행하기 때문에 비교적 작은 크기의 코드로 구현 할 수 있다는 장점이 있다. 이러한 장점들은, 다른 운용모드들에 비해 AES-CCM의 융통성 있는 적용을 가능하게 한다.

[0008] 그러나, 기존 기술은, 암호화 및 메시지 무결성을 위해 두 가지 작업을 필요로 하는 AES-CCM을 처리함에 있어 단일 코어를 이용하는 점에서, 암호화 프로세스 및 컴퓨팅 시간이 지연되고 처리 속도가 떨어지는 문제점이 있다.

[0009] 또한, 기존 기술은, AES-CCM 처리 속도를 고려하여 전용 하드웨어 칩을 사용하는 점에서, 자원 효율성이 떨어지며 호환성을 저해하는 문제점이 있다.

발명의 내용

해결 하고자하는 과제

[0010] 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위하여 창안된 것으로, AES-CCM의 계산 특성과 멀티코어 프로세서의 동기화 오버헤드를 고려하여 데이터 수준 병렬화와 태스크 수준 병렬화를 동시에 활용함으로써, 수행시간을 효과적으로 감소시킬 수 있는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법을 제공하는 것을 그 목적으로 한다.

[0011] 또한, 멀티코어 프로세서를 이용하여 AES-CCM를 병렬처리함으로써, 데이터 기밀성 및 무결성을 동시에 보장함은 물론, 처리 속도 및 자원 효율성을 개선하고 프로세서 호환성을 지니는 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법을 제공하는 것을 그 목적으로 한다.

과제 해결수단

[0012] 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치를 이루는 구성수단은, AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당부, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어들에 의하여 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당하되, 상기 CTR 병렬처리 및 CBC-MAC 병렬 처리를 순차적으로 수행시키는 데이터 수준 병렬화부, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 멀티코어 프로세서를 포함하여 이루어진 것을 특징으로 한다.

[0013] 또한, 상기 코어 할당부는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당하고, 상기 CBC-MAC을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당하는 것을 특징으로 한다.

[0014] 여기서, 상기 코어 할당부는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 것을 특징으로 한다.

[0015] 여기서, 상기 코어 할당부는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 것을 특징으로 한다.

[0016] 본 발명의 다른 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치를 이루는 구성수단은, AES-CCM의 처리 대상 태스크인 CTR를 병렬처리 하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당부, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어(들)에 의하여 순차 또는 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 할당하되, 상기 CTR 병렬 처리와 상기 CBC-MAC 순차 또는 병렬 처리를 병행적으로 수행시키는 데이터 및 태스크 수준 병렬화부, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 멀티코어 프로세서를 포함하여 이루어진 것을 특징으로 한다.

[0017] 또한, 상기 코어 할당부는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당하고, 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 나머지 코어들에서 선택하여 할당하되, 순차 처리인 경우에는 1개의 코어를 할당하고 병렬 처리인 경우에는 2개 이상의 코어들을 할당하는 것을 특징으로 한다.

[0018] 여기서, 상기 코어 할당부는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 것을 특징으로 한다.

[0019] 여기서, 상기 코어 할당부는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 것을 특징으로 한다.

[0020] 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법을 이루는 구성수단은, 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계, 상기 컴퓨터 시스템에서, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어들에 의하여 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당하되, 상기 CTR 병렬 처리 및 CBC-MAC 병렬 처리가 순차적으로 수행되도록 지시하는 데이터 수준 병렬화 단계, 상기 멀티코어 프로세서에서, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 순차적으로 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 단계를 포함하여 이루어진 것을 특징으로 한다.

[0021] 또한, 상기 코어 할당 단계는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당하고, 상기 CBC-MAC을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당하

는 단계인 것을 특징으로 한다.

[0022] 여기서, 상기 코어 할당 단계는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 단계인 것을 특징으로 한다.

[0023] 여기서, 상기 코어 할당 단계는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 단계인 것을 특징으로 한다.

[0024] 본 발명의 다른 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 방법을 이루는 구성수단은, 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, AES-CCM의 처리 대상 태스크인 CTR를 병렬처리 하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계, 상기 컴퓨터 시스템에서, 상기 처리 대상 태스크인 CTR이 할당된 코어들에 의하여 병렬처리되도록, 데이터 블록들을 상기 할당된 코어들에 각각 할당하고, 상기 처리 대상 태스크인 CBC-MAC이 할당된 코어(들)에 의하여 순차 또는 병렬처리되도록, 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 할당하되, 상기 CTR 병렬 처리와 상기 CBC-MAC 순차 또는 병렬 처리가 병행적으로 수행되도록 지시하는 데이터 및 태스크 수준 병렬화 단계, 상기 멀티코어 프로레서에서, 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 할당된 코어들을 통해 각각 병렬처리하되, 상기 CBC-MAC을 병렬처리하는 경우, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하는 단계를 포함하여 이루어진 것을 특징으로 한다.

[0025] 또한, 상기 코어 할당 단계는 상기 CTR을 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당하고, 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 상기 멀티코어 프로세서의 코어들 중, 나머지 코어들에서 선택하여 할당하되, 순차 처리인 경우에는 1개의 코어를 할당하고 병렬 처리인 경우에는 2개 이상의 코어들을 할당하는 단계인 것을 특징으로 한다.

[0026] 여기서, 상기 코어 할당 단계는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정하는 단계인 것을 특징으로 한다.

[0027] 여기서, 상기 코어 할당 단계는 상기 CBC-MAC을 병렬처리하기 위하여 상기 블록내 계산 항목들마다 다른 코어 수를 할당할 수 있고, 블록내 계산 항목들 각각에 할당되는 코어 수를 처리량과 동기화 오버헤드를 고려하여 결정하는 단계인 것을 특징으로 한다.

효 과

[0028] 상기와 같은 과제 및 해결수단을 가지는 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법에 의하면, AES-CCM의 계산 특성과 멀티코어 프로세서의 동기화 오버헤드를 고려하여 데이터 수준 병렬화와 태스크 수준 병렬화를 동시에 활용함으로써, 수행시간을 효과적으로 감소시킬 수 있는 장점이 있다.

[0029] 또한, 멀티코어 프로세서를 이용하여 AES-CCM를 병렬처리함으로써, 데이터 기밀성 및 무결성을 동시에 보장함은 물론, 처리 속도 및 자원 효율성을 개선하고 프로세서 호환성을 지니는 장점이 있다.

발명의 실시를 위한 구체적인 내용

[0030] 이하, 첨부된 도면을 참조하여 상기와 같은 과제, 해결수단 및 효과를 가지는 본 발명인 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치 및 그 방법에 관한 바람직한 실시예를 상세하게 설명한다.

[0031] 그런데, 본 발명이 AES-CCM 병렬 처리에 관한 것이기 때문에, 본 발명의 이해를 돕기 위하여 AES-CCM의 처리에 관한 기본적인 사항들을 소개한다.

[0032] 우선, 일반적인 AES-CCM 병렬처리 장치가 수행하는 AES-CCM 처리 과정을 분석한다. 도 1에는 AES-CCM의 처리 과정이 순서도로 도시되어 있다.

[0033] 도 1에 도시된 바와 같이, AES-CCM의 처리 과정은, 키 초기화 과정(S210), 메시지 헤더 검증 처리 과정(S220), CTR 및 CBC-MAC을 이용하여 메시지 암호화/인증을 수행하는 과정(S230), 및 MAC 계산을 통해 MAC 값을 검증하는 과정(S240)을 포함한다.

[0034] 상기 키 초기화 과정(S210) 및 메시지의 헤더 검증 처리 과정(S220)의 경우, 작은 데이터를 처리하기 때문에 전체 수행시간에 큰 영향을 미치지 않는다. 또한, 상기 MAC 계산 과정(S240)도 암호화 처리가 된 블록을 XOR 연산만 하는 과정이기 때문에 굳이 병렬처리할 이유가 없다. 실제 구현에 있어서, 데이터 처리량이 적은 태스크를 병렬화하는 경우 오히려 병렬화에 따른 복잡한 처리 과정이 효율성을 저해할 수 있다.

[0035] 한편, 상기 메시지 암호화/인증 과정(S230)은, CTR(Counter)을 이용하여 메시지를 암호화하는 계산과 CBC-MAC(Cipher Block Chaining-Message Authentication Code Protocol)을 이용하여 MAC 블록을 생성하는 계산을 수행한다. 상기 메시지 암호화/인증 과정(S230)은, 전체 메시지를 암호화/인증 블록의 개수만큼 처리해야하기 때문에 AES-CCM 전체 수행시간의 대부분을 차지하게 된다. 또한, 상기 두 과정은 모두 AES 암호화를 이용하고 같은 크기의 메시지를 처리하기 때문에 거의 비슷한 수행 시간이 소요된다.

[0036] 따라서, 본 발명은, AES-CCM에 있어서, 특히 상기 메시지 암호화/인증 과정(S230)을 효율적으로 병렬처리하고자 하는 것이다.

[0037] 도 2에는 CTR 운용모드의 처리 과정이 도시되어 있다.

[0038] 도 2에 도시된 바와 같이, CTR은 임의의 수(Nonce)와 카운터(Counter)를 이용해 만든 블록을 일정한 값으로 증가하여 암호화에 사용하기 때문에 블록 암호를 스트림 암호처럼 사용할 수 있게 해준다. 특히, 키와 Nonce를 미리 계산할 수 있기 때문에 데이터를 실시간 암호화할 수 있으며, 처리 과정에서 데이터 간에 종속성을 지니지 않는다. 아래에서 다시 설명하겠지만, 본 발명에 따른 병렬처리 장치 및 방법은, 상기 CTR의 데이터 독립성을 고려하여 상기 CTR에 대해 데이터 병렬화(data-level parallelism 또는 data parallelism)를 수행한다.

[0039] 도 3에는 CBC-MAC 운용모드의 처리 과정이 도시되어 있다.

[0040] 도 3에 도시된 바와 같이, 무결성을 보장하는 방법인 CBC-MAC는, 한 메시지 블록을 초기 벡터(IV)와 같이 암호화하는 것을 시작으로 그 값을 다음 암호화에 이용하는 체이닝(Chaining) 기법인 CBC(Cipher Block Chaining)를 이용한다. CBC-MAC은, CBC의 마지막 암호화된 블록을 메시지의 무결성을 인증하는 코드인 MAC으로 사용하는 방법이다.

[0041] CBC는 블록 암호화 데이터의 독립성을 제거하는 체이닝을 수행하기 때문에 일부 데이터가 변조되었을 때 나머지 데이터들에게도 변조를 전파하는 특성을 가진다. 즉, CBC의 마지막 블록을 MAC으로 사용하여 변조를 확인하면 데이터의 무결성을 검증할 수 있다.

[0042] 도 4에는 AES-CCM 운용모드의 전체 처리 과정이 도시되어 있다.

[0043] 도 4에 도시된 바와 같이, AES-CCM은 하나의 키(key)로 기밀성과 무결성을 동시에 보장할 수 있다. 즉, AES-CCM은 운용모드 중 하나인 상기 CTR을 이용하여 데이터의 기밀성을 제공하면서, 상기 CBC-MAC을 이용하여 메시지 인증 및 무결성을 제공한다. 먼저, CBC를 이용하여 MAC으로 사용할 암호화 블록을 만들고, CTR을 이용해 메시지를 암호화한다. 실시예에 따라, 상기 CBC 및 상기 CTR의 처리 순서는 조정될 수 있다.

[0044] 이하에서는 상기와 같은 절차로 처리되는 AES-CCM을 병렬처리를 통한 처리 속도를 향상시킬 수 있는 장치 및 방법에 대하여 설명한다. 본 발명은 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서 구현될 수 있다. 한편, 데이터 종속성이 있는 CBC-MAC 태스크를 처리하기 위하여, 동기화를 삽입하고, 이 동기화의 오버헤드를 고려하여 병렬처리를 위한 최적의 코어 개수로 상기 AES-CCM이 처리되는 방안에 대하여 설명한다.

[0045] 도 5는 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치의 구성도이다.

[0046] 도 5에 도시된 바와 같이, 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 코어 할당부(10), 데이터 수준 병렬화부(20) 및 멀티코어 프로세서(30)을 포함하여 이루어진다.

[0047] 상기 코어 할당부(10)는 AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당한다. 즉, 상기 코어 할당부(10)는 멀티코어 프로세서를 구성하는 복수개의 코어들 중에, AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 코어들을 할당하고, 동시에 AES-CCM의 처리 대상 태스크인 CBC-MAC을 병렬 처리하기 위한 코어들을 할당한다.

[0048] 후술하겠지만, 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 CTR 병렬처리와 CBC-MAC 병렬 처리를 순차적으로 처리한다. 즉, CTR 병렬 처리를 먼저 수행한 후, CBC-MAC 병렬 처리를 수행하거나, CBC-MAC 병렬 처리를 먼저 수행한 후, CTR 병렬 처리를 수행한다. 이와 같은 순차적 처리 순서는

사전에 설정되어 있거나, 또는 사용자에 의하여 조정될 수 있도록 구성된다.

[0049] 이와 같이, 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 CTR 병렬 처리와 CBC-MAC 병렬 처리를 순차적으로 처리하기 때문에, 각 병렬 처리(CTR 병렬 처리 또는 CBC-MAC 병렬 처리)에 할당될 최대 코어 수는 상기 멀티코어 프로세서를 구성하는 코어 수가 될 것이다.

[0050] 후술하겠지만, 상기 코어 할당부(10)는 상기 CTR 병렬 처리 또는 CBC-MAC 병렬 처리를 수행할 코어를 할당할 때는 동기화 오버헤드, CBC-MAC 처리 시간, CBC-MAC 처리 블록내의 계산 항목들의 처리량 등을 고려하여 최적의 코어 수를 할당한다. 이와 같이, 할당될 코어 수는 사전 실험 결과를 통하여 처리 시간을 최소화하는 코어 수로 결정될 수 있다. 즉, 사전에 할당될 코어 수가 결정될 수 있다. 다만 사용자의 조정에 의하여 할당될 코어 수는 변경될 수 있다.

[0051] 상기와 같이, 코어 할당부(10)에 의하여 AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들이 할당되면, 상기 데이터 수준 병렬화부(20)는 상기 코어 할당부(10)가 할당한 코어들에 처리할 데이터 블록들을 할당하거나, 블록내 계산 항목들 각각의 계산을 할당한다.

[0052] 즉, 상기 데이터 수준 병렬화부(20)는 상기 처리 대상 태스크인 CTR이 상기 코어 할당부(10)에 의하여 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 데이터 블록들을 상기 할당된 코어들에게 각각 할당한다. 또한, 상기 처리 대상 태스크인 CBC-MAC이 상기 코어 할당부(10)에 의하여 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당한다.

[0053] 이와 같이, 할당된 코어들에게 데이터 블록(CRT 병렬 처리를 위한 데이터 블록) 또는 블록내 계산 항목들 각각의 계산(CBC-MAC 병렬 처리를 위한 계산 항목들 각각의 계산)을 할당한 상기 데이터 수준 병렬화부(20)는 상기 CTR 병렬처리 및 CBC-MAC 병렬 처리를 순차적으로 수행시킨다.

[0054] 즉, 상기 데이터 수준 병렬화부(20)는 상기 CRT와 CBC-MAC 각각이 데이터 수준에서 병렬 처리될 수 있도록 지시하고, 태스크 수준에서 상기 CRT와 CBC-MAC이 동시에 병렬처리하지 않는다. 따라서, 상기 CRT 병렬 처리와 CBC-MAC 병렬 처리는 순차적으로 처리되는 것이다.

[0055] 한편, 전체 수행 시간의 대부분을 차지하는 메시지 암호화/인증에서(CRT 처리 및 CBC-MAC 처리에서), 데이터 수준 병렬화(data-level parallelism) 처리를 활용하기 위해서는 블록 데이터 사이의 종속성이 없어야 한다. 그러나, CTR 모드와는 달리 CBC-MAC의 경우 블록간의 체이닝(chainnig)으로 인하여 종속성이 존재하다.

[0056] 따라서, 정확한 계산을 담보하면서 CBC-MAC을 병렬 처리하기 위해서는 블록들을 병렬처리해서는 안되고, 블록내 계산 항목들 각각의 계산을 병렬처리하면서, 동기화를 반드시 삽입해야 한다. 반면, 상기 CTR의 경우에는 블록 데이터 사이의 종속성이 없기 때문에 블록들을 할당된 코어에 배분하여 병렬처리할 수 있다.

[0057] 상기 CBC-MAC 처리에 있어서, 블록내 계산 항목들은 바이트 단위에서의 변환함수를 통한 계산처리를 수행하는 SubByte, 행 변환 함수를 통한 계산처리를 수행하는 ShiftRow, 열 변환 함수를 통한 계산처리를 수행하는 MixColumn, 합성 함수를 통한 계산처리를 수행하는 AddRoundKey가 있다.

[0058] 따라서, 상기 데이터 수준 병렬화부(20)는 상기 코어 할당부(10)에 의하여 상기 CBC-MAC에 할당된 코어들을 통하여 병렬처리가 수행되도록 하기 위해서, 상기 할당된 코어들에게 블록내 계산 항목들 각각의 계산을 할당하는 것이다.

[0059] 상기와 같이, 데이터 수준 병렬화부(20)에 의하여 데이터 블록 또는 블록내 계산 항목들 각각의 계산이 할당되면, 상기 데이터 수준 병렬화부(20)는 멀티코어 프로세서의 코어들로 하여금 병렬처리가 수행되도록 제어한다.

[0060] 즉, 상기 데이터 수준 병렬화부(20)가 CTR 태스크의 병렬 처리를 먼저 수행하는 것으로 지시하면, 상기 CTR 태스크 처리를 위해 할당된 코어들은 상기 데이터 수준 병렬화부(20)에 의하여 할당된 데이터 블록들을 병렬처리한다. CTR 병렬 처리가 완료되면, 상기 데이터 수준 병렬화부(20)는 CBC-MAC 태스크의 병렬 처리 지시를 하고, 상기 CBC-MAC 태스크 처리를 위해 할당된 코어들은 상기 데이터 수준 병렬화부(20)에 의하여 할당된 블록내 계산 항목들 각각의 계산을 처리한다. 이때, 상기 CBC-MAC 병렬처리를 수행하는 코어들 사이에는 반드시 동기화가 삽입되어 병렬처리된다.

[0061] 정리하면, 상기 멀티코어 프로세서(30)는 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 코어 할당부(10)에 의하여 할당된 코어들을 통해 각각 병렬처리한다. 다만, 상기 CTR 병렬 처리를 수행하는 코어들은 상기 데이터 수준 병렬화부(20)에 의하여 할당된 데이터 블록들을 병렬 처리하고, 상기 CBC-MAC을 병렬처리하는 경우에는,

블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행한다.

[0062] 한편, 상기 멀티코어 프로세서의 코어 수는 CTR 및 CBC-MAC을 각각 데이터 수준에서 병렬처리함과 동시에, 상기 CTR 병렬처리와 CBC-MAC 병렬처리를 순차적으로 처리하기 때문에, 최소 듀얼 코어(2 코어)이면 적용 가능하다. 따라서, 상기 멀티코어 프로세서의 코어 수는 2개 이상으로 구성되면, 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리를 수행할 수 있다.

[0063] 상기와 같은 코어 수(2 코어 이상)를 가지는 멀티코어 프로세서의 코어들을 상기 CTR 및 CBC-MAC 병렬 처리에 할당하는 상기 코어 할당부(10)는 상기 CTR을 병렬 처리하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당한다. 즉, 상기 CTR 병렬처리의 속도를 최대한 빠르게 하기 위하여, 멀티코어 프로세서의 모든 코어들을 할당한다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면 N개의 코어 모두를 상기 CTR 병렬 처리를 위하여 할당한다.

[0064] 반면, 상기 코어 할당부(10)는 상기 CBC-MAC을 병렬처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당한다. 즉, 상기 코어 할당부(20)는 멀티코어 프로세서의 코어 모두를 상기 CBC-MAC 병렬처리를 위하여 할당할 수도 있고, 일부의 코어를 할당할 수도 있으며, 최소 코어 수인 2개의 코어만 할당할 수도 있다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면, N개의 코어 모두를 상기 CBC-MAC 병렬 처리를 위하여 할당할 수도 있고, N-1개의 코어만을 할당할 수도 있다. 다만, 제1 실시예에 적용되는 최소 코어 수는 2 코어이기 때문에, 최소 2코어 이상이어야 한다. 즉, 할당될 코어 수는 2보다 크거나 같고, N보다 작거나 같은 수이다.

[0065] 이와 같이, CBC-MAC에 할당되는 코어 수가 멀티코어 프로세서의 모든 코어 수가 아닌 이유는 실제 멀티코어 프로세서에서 코어 간 동기화 오버헤드가 상당하고 코어의 개수에 따라 일반적으로 오버헤드가 증가하기 때문이다. 즉, 동기화 오버헤드를 고려하여, 상기 CBC-MAC 병렬 처리를 위한 코어 수를 결정해야 한다.

[0066] 구체적으로 설명하면, 상기 코어 할당부(10)는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정한다. 즉, 병렬 처리를 위한 코어 수가 많으면, 동기화 오버헤드가 고려되기 때문에 처리 속도가 항상 빠르지 않는다. 따라서, 동기와 오버헤드와 CBC-MAC 처리 시간의 상관관계를 따져서 최적의 코어 수를 결정해야 한다.

[0067] 이와 같은, 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부(10)에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0068] 한편, 상기 코어 할당부(10)는 상기 CBC-MAC을 병렬처리 하기 위하여 상기 블록 내 계산 항목들마다 다른 코어 수를 할당할 수도 있다. 이 때, 상기 코어 할당부(10)는 상기 블록내 계산 항목들 각각에 할당되는 코어 수를 각 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 결정한다.

[0069] 즉, 계산 항목들의 처리량이 많다고 하여, 무조건 많은 코어들을 할당하게 되면, 동기화 오버헤드가 커지는 문제점이 발생한다. 따라서, 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 최적의 코어 수를 결정한다. 예를 들어, SubBytes는 2-코어, ShiftRows는 1-코어, MixColumns는 2-코어, AddRoundKey는 3-코어로 할당하는 것과 같이, 각 계산 항목들마다 다른 코어 수를 할당할 수 있다.

[0070] 이와 같은, 각 계산 항목들에 할당될 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부(10)에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0071] 이상에서 설명한 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치에 의하여 CTR 병렬 처리 및 CBC-MAC 병렬처리를 순차적으로 수행하면, 처리 속도가 개선된다. 왜냐하면, CBC-MAC 병렬 처리를 수행할 코어 수를 동기화 오버헤드, CBC-MAC 처리 시간 및 각 계산 항목들의 처리량 등을 고려하여 결정하

여, 최적의 코어 수를 활용하기 때문이다.

[0072] 다음은, 상기와 같은 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치에 의한 병렬 처리 방법에 대하여 설명한다.

[0073] 먼저, 본 발명에 따른 병렬 처리 방법은 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서 처리될 수 있다. 따라서, 멀티코어 프로세서가 사용되는 모든 컴퓨터 시스템에서 본 발명은 적용될 수 있다.

[0074] 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서 AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계를 수행한다.

[0075] 즉, 상기 코어 할당 단계에서 멀티코어 프로세서를 구성하는 복수개의 코어들 중에, AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 코어들을 할당하고, 동시에 AES-CCM의 처리 대상 태스크인 CBC-MAC을 병렬 처리하기 위한 코어들을 할당한다.

[0076] 후술하겠지만, 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 방법에서는 CTR 병렬 처리와 CBC-MAC 병렬 처리를 순차적으로 처리한다. 즉, CTR 병렬 처리를 먼저 수행한 후, CBC-MAC 병렬 처리를 수행하거나, CBC-MAC 병렬 처리를 먼저 수행한 후, CTR 병렬 처리를 수행한다. 이와 같은 순차적 처리 순서는 사전에 설정되어 있거나, 또는 사용자에 의하여 조정될 수 있도록 구성된다.

[0077] 이와 같이, 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치에서 병렬 처리를 수행하는 방법은 CTR 병렬 처리와 CBC-MAC 병렬 처리를 순차적으로 처리하기 때문에, 각 병렬 처리(CTR 병렬 처리 또는 CBC-MAC 병렬 처리)에 할당될 최대 코어 수는 상기 멀티코어 프로세서를 구성하는 코어 수가 될 것이다.

[0078] 후술하겠지만, 상기 코어 할당 단계에서는 상기 CTR 병렬 처리 또는 CBC-MAC 병렬 처리를 수행할 코어를 할당할 때는 동기화 오버헤드, CBC-MAC 처리 시간, CBC-MAC 처리 블록내의 계산 항목들의 처리량 등을 고려하여 최적의 코어 수를 할당한다. 이와 같이, 할당될 코어 수는 사전 실험 결과를 통하여 처리 시간을 최소화하는 코어 수로 결정될 수 있다. 즉, 사전에 할당될 코어 수가 결정될 수 있다. 다만 사용자의 조정에 의하여 할당될 코어 수는 변경될 수 있다.

[0079] 상기와 같이, 코어 할당 단계에서 AES-CCM의 처리 대상 태스크인 CTR 및 CBC-MAC을 각각 병렬처리 하기 위한 멀티코어 프로세서의 코어들이 할당되면, 상기 컴퓨터 시스템에서 데이터 수준 병렬화 단계가 수행된다. 이 데이터 수준 병렬화 단계에서는 상기 코어 할당 단계에서 할당한 코어들에 처리할 데이터 블록들을 할당하거나, 블록내 계산 항목들 각각의 계산을 할당한다.

[0080] 즉, 상기 데이터 수준 병렬화 단계에서는 상기 처리 대상 태스크인 CTR이 상기 코어 할당 단계에서 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 데이터 블록들을 상기 할당된 코어들에게 각각 할당한다. 또한, 상기 처리 대상 태스크인 CBC-MAC이 상기 코어 할당 단계에서 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 블록내 계산 항목들 각각의 계산을 상기 할당된 코어들에 각각 할당한다.

[0081] 이와 같이, 할당된 코어들에게 데이터 블록(CRT 병렬 처리를 위한 데이터 블록) 또는 블록내 계산 항목들 각각의 계산(CBC-MAC 병렬 처리를 위한 계산 항목들 각각의 계산)을 할당한 상기 데이터 수준 병렬화 단계에서는 부가적으로 상기 CTR 병렬처리 및 CBC-MAC 병렬 처리가 순차적으로 수행되도록 지시하는 단계를 포함한다.

[0082] 즉, 상기 데이터 수준 병렬화 단계에서는 상기 CRT와 CBC-MAC 각각이 데이터 수준에서 병렬 처리될 수 있도록 지시하고, 태스크 수준에서 상기 CRT와 CBC-MAC이 동시에 병렬처리하지 않는다. 따라서, 상기 CRT 병렬 처리와 CBC-MAC 병렬 처리는 순차적으로 처리되는 것이다.

[0083] 한편, 전체 수행 시간의 대부분을 차지하는 메시지 암호화/인증에서(CRT 처리 및 CBC-MAC 처리에서), 데이터 수준 병렬화(data-level parallelism) 처리를 활용하기 위해서는 블록 데이터 사이의 종속성이 없어야 한다. 그러나, CTR 모드와는 달리 CBC-MAC의 경우 블록간의 체이닝(chainnig)으로 인하여 종속성이 존재하다.

[0084] 따라서, 정확한 계산을 담보하면서 CBC-MAC을 병렬 처리하기 위해서는 블록들을 병렬처리해서는 안되고, 블록내 계산 항목들 각각의 계산을 병렬처리하면서, 동기화를 반드시 삽입해야 한다. 반면, 상기 CTR의 경우에는 블록 데이터 사이의 종속성이 없기 때문에 블록들을 할당된 코어에 배분하여 병렬처리할 수 있다.

[0085] 상기 CBC-MAC 처리에 있어서, 블록내 계산 항목들은 바이트 단위에서의 변환함수를 통한 계산처리를 수행하는

SubByte, 행 변환 함수를 통한 계산처리를 수행하는 ShiftRow, 열 변환 함수를 통한 계산처리를 수행하는 MixColumn, 합성 함수를 통한 계산처리를 수행하는 AddRoundKey가 있다.

[0086] 따라서, 상기 데이터 수준 병렬화 단계에서는 상기 코어 할당 단계에서 상기 CBC-MAC에 할당된 코어들을 통하여 병렬처리가 수행되도록 하기 위해서, 상기 할당된 코어들에게 블록내 계산 항목들 각각의 계산을 할당하는 것이다.

[0087] 상기와 같이, 데이터 수준 병렬화 단계에서 데이터 블록 또는 블록내 계산 항목들 각각의 계산이 할당되면, 상기 데이터 수준 병렬화 단계에서는 추가적으로 멀티코어 프로세서, 코어들로 하여금 병렬처리가 수행되도록 제어한다.

[0088] 즉, 상기 데이터 수준 병렬화 단계에서 CTR 태스크의 병렬 처리를 먼저 수행하는 것으로 지시하면, 상기 CTR 태스크 처리를 위해 할당된 코어들은 상기 데이터 수준 병렬화 단계에서 할당된 데이터 블록들을 병렬처리한다. CTR 병렬 처리가 완료되면, 상기 데이터 수준 병렬화 단계에서는 CBC-MAC 태스크의 병렬 처리 지시를 하고, 상기 CBC-MAC 태스크 처리를 위해 할당된 코어들은 상기 데이터 수준 병렬화 단계에서 할당된 블록내 계산 항목들 각각의 계산을 처리한다. 이때, 상기 CBC-MAC 병렬처리를 수행하는 코어들 사이에는 반드시 동기화가 삽입되어 병렬처리된다.

[0089] 정리하면, 상기 멀티코어 프로세서에서는 상기 처리 대상 태스크인 CTR 및 CBC-MAC을 상기 코어 할당 단계에서 할당된 코어들을 통해 각각 병렬처리한다. 다만, 상기 CTR 병렬 처리를 수행하는 코어들은 상기 데이터 수준 병렬화 단계에서 할당된 데이터 블록들을 병렬 처리하고, 상기 CBC-MAC을 병렬처리하는 경우에는, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행한다.

[0090] 한편, 상기 멀티코어 프로세서의 코어 수는 CTR 및 CBC-MAC을 각각 데이터 수준에서 병렬처리함과 동시에, 상기 CTR 병렬처리와 CBC-MAC 병렬처리를 순차적으로 처리하기 때문에, 최소 듀얼 코어(2 코어)이면 적용 가능하다. 따라서, 상기 멀티코어 프로세서의 코어 수는 2개 이상으로 구성되면, 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리를 수행할 수 있다.

[0091] 상기와 같은 코어 수(2 코어 이상)를 가지는 멀티코어 프로세서의 코어들을 상기 CTR 및 CBC-MAC 병렬 처리에 할당하는 상기 코어 할당 단계는 상기 CTR을 병렬 처리하기 위하여 상기 멀티코어 프로세서의 모든 코어들을 할당한다. 즉, 상기 CTR 병렬처리의 속도를 최대한 빠르게 하기 위하여, 멀티코어 프로세서의 모든 코어들을 할당한다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면 N개의 코어 모두를 상기 CTR 병렬 처리를 위하여 할당한다.

[0092] 반면, 상기 코어 할당 단계에서는 상기 CBC-MAC을 병렬처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 2개 이상의 코어들을 할당한다. 즉, 상기 코어 할당부(20)는 멀티코어 프로세서의 코어 모두를 상기 CBC-MAC 병렬처리를 위하여 할당할 수도 있고, 일부의 코어를 할당할 수도 있으며, 최소 코어 수인 2개의 코어만 할당할 수도 있다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면, N개의 코어 모두를 상기 CBC-MAC 병렬 처리를 위하여 할당할 수도 있고, N-1개의 코어만을 할당할 수도 있다. 다만, 제1 실시예에 적용되는 최소 코어 수는 2 코어이기 때문에, 최소 2코어 이상이어야 한다. 즉, 할당될 코어 수는 2보다 크거나 같고, N보다 작거나 같은 수이다.

[0093] 이와 같이, CBC-MAC에 할당되는 코어 수가 멀티코어 프로세서의 모든 코어 수가 아닌 이유는 실제 멀티코어 프로세서에서 코어 간 동기화 오버헤드가 상당하고 코어의 개수에 따라 일반적으로 오버헤드가 증가하기 때문이다. 즉, 동기화 오버헤드를 고려하여, 상기 CBC-MAC 병렬 처리를 위한 코어 수를 결정해야 한다.

[0094] 구체적으로 설명하면, 상기 코어 할당 단계에서는 상기 CBC-MAC을 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정한다. 즉, 병렬 처리를 위한 코어 수가 많으면, 동기화 오버헤드가 고려되기 때문에 처리 속도가 항상 빠르지 않는다. 따라서, 동기와 오버헤드와 CBC-MAC 처리 시간의 상관관계를 따져서 최적의 코어 수를 결정해야 한다.

[0095] 이와 같은, 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여

사용할 수도 있다.

[0096] 한편, 상기 코어 할당 단계에서는 상기 CBC-MAC을 병렬처리 하기 위하여 상기 블록 내 계산 항목들마다 다른 코어 수를 할당할 수도 있다. 이 때, 상기 코어 할당 단계에서는 상기 블록내 계산 항목들 각각에 할당되는 코어 수를 각 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 결정한다.

[0097] 즉, 계산 항목들의 처리량이 많다고 하여, 무조건 많은 코어들을 할당하게 되면, 동기화 오버헤드가 커지는 문제점이 발생한다. 따라서, 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 최적의 코어 수를 결정한다. 예를 들어, SubBytes는 2-코어, ShiftRows는 1-코어, MixColumns는 2-코어, AddRoundKey는 3-코어로 할당하는 것과 같이, 각 계산 항목들마다 다른 코어 수를 할당할 수 있다.

[0098] 이와 같은, 각 계산 항목들에 할당될 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 코어 할당부에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0099] 이상에서 설명한 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치를 통한 병렬처리 방법에 의하여 CTR 병렬 처리 및 CBC-MAC 병렬처리를 순차적으로 수행하면, 처리 속도가 개선된다. 왜냐하면, CBC-MAC 병렬 처리를 수행할 코어 수를 동기화 오버헤드, CBC-MAC 처리 시간 및 각 계산 항목들의 처리량 등을 고려하여 결정하여, 최적의 코어 수를 활용하기 때문이다.

[0100] 이상에서 설명한 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 방법의 구체적인 예시를 첨부된 도면을 참조하여 살펴본다.

[0101] 도 7 내지 도 9는 본 발명의 제1 실시예인 데이터 수준 병렬화을 이용하여 AES-CCM 병렬 처리 방법의 예시를 보여준다. 한편, 도 7 내지 도 9의 예시는 4-코어를 이용하여 데이터 수준의 병렬화를 이용하여 AES-CCM 병렬 처리를 수행한 예시이다.

[0102] 전체 수행시간의 대부분을 차지하는 메시지 암호화/인증에서 data-level parallelism을 활용하기 위해서는 블록 데이터 사이의 종속성이 없어야 한다. 그러나 CTR 모드와 달리, CBC-MAC의 경우 블록간의 체이닝으로 종속성이 존재하여 정확한 계산을 위해서는 코어간 동기화를 삽입하여야 한다. 즉, 도 7과 같이 4-코어를 활용하여 CTR을 병렬처리하고, 블록 데이터간 종속성이 있는 CBC-MAC의 경우 블록내 계산(즉, AES 계산중 SubBytes/ShiftRows/MixColumns/AddRoundKey)을 4-코어에서 병렬처리하기 위하여 동기화가 삽입되는 병렬 처리 방법을 적용할 수 있다.

[0103] 그러나, 실제 멀티코어 프로세서에서 코어간 동기화 오버헤드가 상당하고 코어의 개수에 따라 증가하기 때문에, 동기화 오버헤드를 고려한 메시지 인증 부분에 할당될 최적 코어 수를 도출할 필요가 있다. 예를 들어, 동기화 오버헤드와 CBC-MAC 계산의 상관관계에 의하여 CBC-MAC 부분을 2개의 코어에서 수행하는 부하 분산 방법을 도 8에 나타내었다.

[0104] 마지막으로, CBC-MAC의 블록내 계산인 SubBytes, ShiftRows, MixColumns, AddRoundKey 각각은 서로 다른 처리량을 요구하므로, 처리량과 동기화 오버 헤드를 고려하여 각각의 계산에 최적인 코어를 다르게 설정하는 병렬처리 방법도 적용할 수 있다. 예를 들어, 도 9에 도시된 바와 같이, SubBytes는 2-코어, ShiftRows는 1-코어, MixColumns는 2-코어, AddRoundKey는 3-코어로 할당된 예를 보여주고 있다.

[0105] 다음은 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치에 관한 바람직한 실시예를 설명한다.

[0106] 앞서 설명한 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치 및 방법은 데이터 수준에서 병렬처리하는 장치 및 방법이다. 그러나, 다음에서 설명할 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치 및 방법은 데이터 수준에서 병렬처리하는 방법(CTR 및 CBC-MAC 모두 데이터 병렬 처리)뿐만 아니라, 태스크 수준에서 병렬처리하는 방법(CTR 처리와 CBC-MAC 처리를 동시에 병행적으로 처리)도 적용되는 실시예이다.

[0107] 도 6은 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치의 구성도이다.

[0108] 도 6에 도시된 바와 같이, 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 코어 할당부(40), 데이터 및 태스크 수준 병렬화부(50) 및 멀티코어 프로세서(60)을 포함하여 이루어진다.

[0109] 상기 코어 할당부(40)는 AES-CCM의 처리 대상 태스크인 CTR을 병렬처리하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당한다. 즉, 상기 코어 할당부(40)는 멀티코어 프로세서를 구성하는 복수개의 코어들 중에, AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 코어들을 할당하고, 동시에 AES-CCM의 처리 대상 태스크인 CBC-MAC을 순차 또는 병렬 처리하기 위한 코어들을 할당한다. 앞서 설명한 제1 실시예와 달리, 상기 코어 할당부(40)는 CBC-MAC을 병렬처리하기 위하여 코어들을 할당할 뿐 아니라, 순차 처리하기 위하여 코어들을 할당하기도 한다. 왜냐하면, 제2 실시예에서는 CBC-MAC 처리를 CTR 병렬 처리와 동시에 병행적으로 처리하기 때문에(태스크 수준에서 병렬화하기 때문에), 전체 처리 속도를 고려하여, 상기 CBC-MAC 처리를 하나의 코어에서 순차처리하는 것이 최적인 경우가 발생하기 때문이다.

[0110] 후술하겠지만, 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 처리한다(이 점이 제1 실시예에 다르다). 즉, CTR 병렬 처리를 먼저 수행한 후, CBC-MAC 병렬 처리를 수행하거나, CBC-MAC 병렬 처리를 먼저 수행한 후, CTR 병렬 처리를 수행하는 것이 아니라, 동시에 병행적으로 처리를 수행한다.

[0111] 이와 같이, 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치는 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 처리하기 때문에, 동시 병행 처리(CTR 병렬 처리 또는 CBC-MAC 병렬 처리)에 할당될 최소 코어 수는 3이 될 것이다. 그러나, CBC-MAC을 병렬 처리하는 경우도 포함하기 때문에, 최소 코어 수는 4가 되는 것이 더 타당하다.

[0112] 후술하겠지만, 상기 코어 할당부(40)는 상기 CTR 병렬 처리 또는 CBC-MAC 병렬 처리를 수행할 코어를 할당할 때는 동기화 오버헤드, CBC-MAC 처리 시간, CBC-MAC 처리 블록내의 계산 항목들의 처리량 등을 고려하여 최적의 코어 수를 할당한다. 이와 같이, 할당될 코어 수는 사전 실험 결과를 통하여 처리 시간을 최소화하는 코어 수로 결정될 수 있다. 즉, 사전에 할당될 코어 수가 결정될 수 있다. 다만 사용자의 조정에 의하여 할당될 코어 수는 변경될 수 있다.

[0113] 상기와 같이, 코어 할당부(40)에 의하여 AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 멀티코어 프로세서의 코어들을 할당하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어(들)이 할당되면, 상기 데이터 및 태스크 수준 병렬화부(50)는 상기 코어 할당부(40)가 할당한 코어들에 처리할 데이터 블록들을 할당하거나, 블록내 계산 항목들 각각의 계산을 할당한다.

[0114] 즉, 상기 데이터 및 태스크 수준 병렬화부(50)는 상기 처리 대상 태스크인 CTR이 상기 코어 할당부(40)에 의하여 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 데이터 블록들을 상기 할당된 코어들에게 각각 할당한다. 또한, 상기 처리 대상 태스크인 CBC-MAC이 상기 코어 할당부(40)에 의하여 할당된 코어(들)에 의하여 순차 또는 병렬처리될 수 있도록, 처리할 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 각각 할당한다.

[0115] 이와 같이, 할당된 코어들에게 데이터 블록(CRT 병렬 처리를 위한 데이터 블록) 또는 블록내 계산 항목들 각각의 계산(CBC-MAC 순차 또는 병렬 처리를 위한 계산 항목들 각각의 계산)을 할당한 상기 데이터 및 태스크 수준 병렬화부(50)는 상기 CTR 병렬처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 수행시킨다.

[0116] 즉, 상기 데이터 및 태스크 수준 병렬화부(50)는 상기 CRT와 CBC-MAC 각각이 데이터 수준에서 병렬 처리될 수 있도록 지시하거나, 상기 CRT는 데이터 수준에서 병렬 처리하고 상기 CBC-MAC은 단순한 순차처리될 수 있도록 지시하고, 태스크 수준에서 상기 CRT와 CBC-MAC이 동시에 병행적으로 처리하도록 지시한다 따라서, 상기 CRT 병렬 처리와 CBC-MAC 순차 또는 병렬 처리는 동시에 병행적으로 처리되는 것이다(태스크 수준에서 병렬화).

[0117] 한편, 전체 수행 시간의 대부분을 차지하는 메시지 암호화/인증에서(CRT 처리 및 CBC-MAC 처리에서), 데이터 수준 병렬화(data-level parallelism) 처리를 활용하기 위해서는 블록 데이터 사이의 종속성이 없어야 한다. 그러나, CTR 모드와는 달리 CBC-MAC의 경우 블록간의 체이닝(chainnig)으로 인하여 종속성이 존재하다.

[0118] 따라서, 정확한 계산을 담보하면서 CBC-MAC을 병렬 처리하기 위해서는 블록들을 병렬처리해서는 안되고, 블록내 계산 항목들 각각의 계산을 병렬처리하면서, 동기화를 반드시 삽입해야 한다. 반면, 상기 CTR의 경우에는 블록 데이터 사이의 종속성이 없기 때문에 블록들을 할당된 코어에 배분하여 병렬처리할 수 있다. 한편, CBC-MAC을

순차처리하는 경우에는 하나의 코어만 사용하기 때문에 동기화 삽입 문제는 발생하지 않는다.

[0119] 상기 CBC-MAC 처리에 있어서, 블록내 계산 항목들은 바이트 단위에서의 변환함수를 통한 계산처리를 수행하는 SubByte, 행 변환 함수를 통한 계산처리를 수행하는 ShiftRow, 열 변환 함수를 통한 계산처리를 수행하는 MixColumn, 합성 함수를 통한 계산처리를 수행하는 AddRoundKey가 있다.

[0120] 따라서, 상기 데이터 및 태스크 수준 병렬화부(50)는 상기 코어 할당부(40)에 의하여 상기 CBC-MAC에 할당된 코어들을 통하여 병렬처리가 수행되도록 하기 위해서, 상기 할당된 코어들에게 블록내 계산 항목들 각각의 계산을 할당하는 것이다. 한편, 상기 데이터 및 태스크 수분 병렬화부(50)는 상기 코어 할당부(40)에 의하여 상기 CBC-MAC에 할당된 코어를 통하여 순차 처리가 수행되도록 하기 위해서는, 할당된 하나의 코어에 블록들을 모두 할당하면 된다. 즉, 순차 처리되기 때문에, 계산이 잘못될 문제는 발생하지 않고, 동기화 삽입 문제도 역시 발생하지 않는다.

[0121] 상기와 같이, 데이터 및 태스크 수준 병렬화부(50)에 의하여 데이터 블록 또는 블록내 계산 항목들 각각의 계산이 할당되면, 상기 데이터 및 태스크 수준 병렬화부(50)는 멀티코어 프로세서의 코어들로 하여금 병렬처리(CBC-MAC의 경우에는 순차 처리가 해당될 수 있음)가 수행되도록 제어한다.

[0122] 즉, 상기 데이터 및 태스크 수준 병렬화부(50)가 CTR 태스크의 병렬 처리와 CBC-MAC 태스크의 순차 또는 병렬처리를 동시에 병행적으로 수행하는 것을 지시하면, 상기 CTR 태스크 처리를 위해 할당된 코어들은 상기 데이터 및 태스크 수준 병렬화부(50)에 의하여 할당된 데이터 블록들을 병렬처리한다. 그리고, 상기 CTR 병렬 처리와 동시에 병행적으로, 상기 CBC-MAC 태스크 처리를 위해 할당된 코어들은 상기 데이터 및 태스크 수준 병렬화부(50)에 의하여 할당된 블록내 계산 항목들 각각의 계산을 순차 또는 병렬처리한다. 이때, 상기 CBC-MAC가 병렬처리된다면, 병렬처리를 수행하는 코어들 사이에는 반드시 동기화가 삽입되어 병렬처리된다.

[0123] 정리하면, 상기 멀티코어 프로세서(30)는 상기 처리 대상 태스크인 CTR을 상기 코어 할당부(40)에 의하여 할당된 코어들을 통해 병렬처리하고, CBC-MAC을 상기 코어 할당부(40)에 의하여 할당된 코어들을 통해 순처 또는 병렬처리한다. 다만, 상기 CTR 병렬 처리를 수행하는 코어들은 상기 데이터 및 태스크 수준 병렬화부(50)에 의하여 할당된 데이터 블록들을 병렬 처리하고, 상기 CBC-MAC을 병렬처리하는 경우에는, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하며, 상기 CBC-MAC을 순차처리하는 경우에는 순차적으로 블록내 계산 항목들을 순차처리하면서, 데이터 블록들을 순차 처리한다.

[0124] 한편, 상기 멀티코어 프로세서의 코어 수는 CTR을 데이터 수준에서 병렬처리하고, CBC-MAC을 순차 처리 또는 데이터 수준에서 병렬처리함과 동시에, 상기 CTR 병렬처리와 CBC-MAC 순처 또는 병렬처리를 동시에 병행적으로 처리하기 때문에(태스크 수준에서 병렬처리하기 때문에), 최소 3-코어이면 적용 가능하다. 그러나, CBC-MAC 태스크 처리를 병렬처리하는 것을 포함하고 있기 때문에, 최소 4-코어인 것이 바람직하다.

[0125] 따라서, 상기 멀티코어 프로세서의 코어 수는 4개 이상으로 구성되면, 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리를 수행할 수 있다.

[0126] 상기와 같은 코어 수(4 코어 이상)를 가지는 멀티코어 프로세서의 코어들을 상기 CTR 병렬 처리 및 상기 CBC-MAC 순차 또는 병렬 처리에 할당하는 상기 코어 할당부(40)는 상기 CTR을 병렬 처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당한다. 즉, 태스크 수준에서 병렬처리를 해야 하기 때문에, 전체 코어의 일부인 절반의 코어들을 할당한다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면 N/2개의 코어를 상기 CTR 병렬 처리를 위하여 할당한다. 이와 같이 전체 코어의 절반을 CTR 태스크 처리에 할당하는 이유는 암호화 부분과 메시지 인증 부분의 계산 시간이 유사하기 때문이다.

[0127] 반면, 상기 코어 할당부(40)는 상기 CBC-MAC을 순차 또는 병렬처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 상기 전체 코어들 중에서 상기 CTR 태스크 처리에 할당한 코어를 뺀 나머지 코어들에서 선택하여 할당한다. 구체적으로, CBC-MAC의 태스크 처리가 순차 처리인 경우에는 1개의 코어를 할당하고, 병렬 처리인 경우에는 2개 이상의 코어들을 할당한다.

[0128] 즉, 상기 코어 할당부(40)는 멀티코어 프로세서의 코어들 중, CRT 태스크 처리에 할당한 코어를 뺀 나머지 코어들 중에서 병렬처리가 가능한 최소 코어 수인 2개 이상의 코어를 상기 CBC-MAC 병렬처리를 위하여 할당한다. 물론, 순차처리인 경우에는 하나의 코어만 할당하면 된다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면, N/2개의 코어가 상기 CRT 태스크 처리에 할당되었기 때문에, 2개보다 크거나 같고 N/2개보다 작거나 같은 수의 코어를 상기 CBC-MAC 병렬 처리를 위하여 할당할 수도 있다.

[0129] 이와 같이, CBC-MAC에 할당되는 코어 수가 멀티코어 프로세서의 모든 코어에서 CTR 태스크 처리에 할당한 코어 수를 뺀 나머지 수가 아닐 수 있는 이유는 실제 멀티코어 프로세서에서 코어 간 동기화 오버헤드가 상당하고 코어의 개수에 따라 일반적으로 오버헤드가 증가하기 때문이다. 즉, 동기화 오버헤드를 고려하여, 상기 CBC-MAC 병렬 처리를 위한 코어 수를 결정해야 한다.

[0130] 구체적으로 설명하면, 상기 코어 할당부(40)는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정한다. 즉, 병렬 처리를 위한 코어 수가 많으면, 동기화 오버헤드가 고려되기 때문에 처리 속도가 항상 빠르지 않는다. 따라서, 동기와 오버헤드와 CBC-MAC 처리 시간의 상관관계를 따져서 최적의 코어 수를 결정해야 한다.

[0131] 이와 같은, 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 순차 또는 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부(40)에 저장하였다가, CBC-MAC 순차 또는 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 순차 또는 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 순차 또는 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0132] 한편, 상기 코어 할당부(40)는 상기 CBC-MAC을 병렬처리 하기 위하여 상기 블록 내 계산 항목들마다 다른 코어 수를 할당할 수도 있다. 이 때, 상기 코어 할당부(40)는 상기 블록내 계산 항목들 각각에 할당되는 코어 수를 각 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 결정한다.

[0133] 즉, 계산 항목들의 처리량이 많다고 하여, 무조건 많은 코어들을 할당하게 되면, 동기화 오버헤드가 커지는 문제점이 발생한다. 따라서, 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 최적의 코어 수를 결정한다. 예를 들어, SubBytes는 1-코어, ShiftRows는 1-코어, MixColumns는 1-코어, AddRoundKey는 2-코어로 할당하는 것과 같이, 각 계산 항목들마다 다른 코어 수를 할당할 수 있다.

[0134] 이와 같은, 각 계산 항목들에 할당될 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부(40)에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0135] 이상에서 설명한 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치에 의하여 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬처리를 동시에 병행적으로 수행하면, 처리 속도가 개선된다. 왜냐하면, CBC-MAC 순차 또는 병렬 처리를 수행할 코어 수를 동기화 오버헤드, CBC-MAC 처리 시간 및 각 계산 항목들의 처리량 등을 고려하여 결정하여, 최적의 코어 수를 활용하기 때문이다.

[0136] 다음은, 상기와 같은 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치에 의한 병렬 처리 방법에 대하여 설명한다.

[0137] 먼저, 본 발명에 따른 병렬 처리 방법은 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서 처리될 수 있다. 따라서, 멀티코어 프로세서가 사용되는 모든 컴퓨터 시스템에서 본 발명은 적용될 수 있다.

[0138] 멀티코어 프로세서를 사용하는 컴퓨터 시스템에서, 먼저 AES-CCM의 처리 대상 태스크인 CTR을 병렬처리하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어들을 할당하는 코어 할당 단계를 수행한다. 즉, 상기 코어 할당 단계에서는 멀티코어 프로세서를 구성하는 복수개의 코어들 중에, AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 코어들을 할당하고, 동시에 AES-CCM의 처리 대상 태스크인 CBC-MAC을 순차 또는 병렬 처리하기 위한 코어들을 할당한다. 앞서 설명한 제1 실시예와 달리, 상기 코어 할당부(40)는 CBC-MAC을 병렬처리하기 위하여 코어들을 할당할 뿐 아니라, 순차 처리하기 위하여 코어들을 할당하기도 한다. 왜냐하면, 제2 실시예에서는 CBC-MAC 처리를 CTR 병렬 처리와 동시에 병행적으로 처리하기 때문에(태스크 수준에서 병렬화하기 때문에), 전체 처리 속도를 고려하여, 상기 CBC-MAC 처리를 하나의 코어에서 순차처리하는 것이 최적인 경우가 발생하기 때문이다.

[0139] 후술하겠지만, 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치를 이용한 병렬 처리 방법은 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 처리한다(이 점이 제1 실시

예에 다르다). 즉, CTR 병렬 처리를 먼저 수행한 후, CBC-MAC 병렬 처리를 수행하거나, CBC-MAC 병렬 처리를 먼저 수행한 후, CTR 병렬 처리를 수행하는 것이 아니라, 동시에 병행적으로 처리를 수행한다.

[0140] 이와 같이, 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치를 통한 병렬 처리 방법은 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 처리하기 때문에, 동시 병행 처리(CTR 병렬 처리 또는 CBC-MAC 병렬 처리)에 할당될 최소 코어 수는 3이 될 것이다. 그러나, CBC-MAC을 병렬 처리하는 경우도 포함하기 때문에, 최소 코어 수는 4가 되는 것이 더 타당하다.

[0141] 후술하겠지만, 상기 코어 할당 단계에서는 상기 CTR 병렬 처리 또는 CBC-MAC 병렬 처리를 수행할 코어를 할당할 때는 동기화 오버헤드, CBC-MAC 처리 시간, CBC-MAC 처리 블록내의 계산 항목들의 처리량 등을 고려하여 최적의 코어 수를 할당한다. 이와 같이, 할당될 코어 수는 사전 실험 결과를 통하여 처리 시간을 최소화하는 코어 수로 결정될 수 있다. 즉, 사전에 할당될 코어 수가 결정될 수 있다. 다만 사용자의 조정에 의하여 할당될 코어 수는 변경될 수 있다.

[0142] 상기와 같이, 코어 할당 단계에서 AES-CCM의 처리 대상 태스크인 CTR을 병렬 처리하기 위한 멀티코어 프로세서의 코어들을 할당하고, CBC-MAC을 순차 또는 병렬처리 하기 위한 멀티코어 프로세서의 코어(들)이 할당되면, 데이터 및 태스크 병렬화 단계를 수행한다. 상기 데이터 및 태스크 수준 병렬화 단계에서는 상기 코어 할당 단계에서 할당한 코어들에 처리할 데이터 블록들을 할당하거나, 블록내 계산 항목들 각각의 계산을 할당한다.

[0143] 즉, 상기 데이터 및 태스크 수준 병렬화 단계에서는 상기 처리 대상 태스크인 CTR이 상기 코어 할당 단계에서 할당된 코어들에 의하여 병렬처리될 수 있도록, 처리할 데이터 블록들을 상기 할당된 코어들에게 각각 할당한다. 또한, 상기 처리 대상 태스크인 CBC-MAC이 상기 코어 할당 단계에서 할당된 코어(들)에 의하여 순차 또는 병렬처리될 수 있도록, 처리할 블록내 계산 항목들 각각의 계산을 상기 할당된 코어(들)에 각각 할당한다.

[0144] 이와 같이, 할당된 코어들에게 데이터 블록(CRT 병렬 처리를 위한 데이터 블록) 또는 블록내 계산 항목들 각각의 계산(CBC-MAC 순차 또는 병렬 처리를 위한 계산 항목들 각각의 계산)을 할당한 상기 데이터 및 태스크 수준 병렬화 단계에서는 상기 CTR 병렬처리와 CBC-MAC 순차 또는 병렬 처리를 동시에 병행적으로 수행시킨다.

[0145] 즉, 상기 데이터 및 태스크 수준 병렬화 단계에서는 상기 CRT와 CBC-MAC 각각이 데이터 수준에서 병렬 처리될 수 있도록 지시하거나, 상기 CRT는 데이터 수준에서 병렬 처리하고 상기 CBC-MAC은 단순한 순차처리될 수 있도록 지시하고, 태스크 수준에서 상기 CRT와 CBC-MAC이 동시에 병행적으로 처리하도록 지시한다 따라서, 상기 CRT 병렬 처리와 CBC-MAC 순차 또는 병렬 처리는 동시에 병행적으로 처리되는 것이다(태스크 수준에서 병렬화).

[0146] 한편, 전체 수행 시간의 대부분을 차지하는 메시지 암호화/인증에서(CRT 처리 및 CBC-MAC 처리에서), 데이터 수준 병렬화(data-level parallelism) 처리를 활용하기 위해서는 블록 데이터 사이의 종속성이 없어야 한다. 그러나, CTR 모드와는 달리 CBC-MAC의 경우 블록간의 체이닝(chainnig)으로 인하여 종속성이 존재하다.

[0147] 따라서, 정확한 계산을 담보하면서 CBC-MAC을 병렬 처리하기 위해서는 블록들을 병렬처리해서는 안되고, 블록내 계산 항목들 각각의 계산을 병렬처리하면서, 동기화를 반드시 삽입해야 한다. 반면, 상기 CTR의 경우에는 블록 데이터 사이의 종속성이 없기 때문에 블록들을 할당된 코어에 배분하여 병렬처리할 수 있다. 한편, CBC-MAC을 순차처리하는 경우에는 하나의 코어만 사용하기 때문에 동기화 삽입 문제는 발생하지 않는다.

[0148] 상기 CBC-MAC 처리에 있어서, 블록내 계산 항목들은 바이트 단위에서의 변환함수를 통한 계산처리를 수행하는 SubByte, 행 변환 함수를 통한 계산처리를 수행하는 ShiftRow, 열 변환 함수를 통한 계산처리를 수행하는 MixColumn, 합성 함수를 통한 계산처리를 수행하는 AddRoundKey가 있다.

[0149] 따라서, 상기 데이터 및 태스크 수준 병렬화 단계에서는 상기 코어 할당 단계에서 상기 CBC-MAC에 할당된 코어들을 통하여 병렬처리가 수행되도록 하기 위해서, 상기 할당된 코어들에게 블록내 계산 항목들 각각의 계산을 할당하는 것이다. 한편, 상기 데이터 및 태스크 수분 병렬화 단계에서는 상기 코어 할당 단계에서 상기 CBC-MAC에 할당된 코어를 통하여 순차 처리가 수행되도록 하기 위해서는, 할당된 하나의 코어에 블록들을 모두 할당하면 된다. 즉, 순차 처리되기 때문에, 계산이 잘못될 문제는 발생하지 않고, 동기화 삽입 문제도 역시 발생하지 않는다.

[0150] 상기와 같이, 데이터 및 태스크 수준 병렬화 단계에서 데이터 블록 또는 블록내 계산 항목들 각각의 계산이 할당되면, 상기 데이터 및 태스크 수준 병렬화 단계에서는 추가적으로 멀티코어 프로세서의 코어들로 하여금 병렬처리(CBC-MAC의 경우에는 순차 처리가 해당될 수 있음)가 수행되도록 제어한다.

[0151] 즉, 상기 데이터 및 태스크 수준 병렬화 단계에서 CTR 태스크의 병렬 처리와 CBC-MAC 태스크의 순차 또는 병렬

처리를 동시에 병행적으로 수행하는 것을 지시하면, 상기 CTR 태스크 처리를 위해 할당된 코어들은 상기 데이터 및 태스크 수준 병렬화 단계에서 할당된 데이터 블록들을 병렬처리한다. 그리고, 상기 CTR 병렬 처리와 동시에 병행적으로, 상기 CBC-MAC 태스크 처리를 위해 할당된 코어들은 상기 데이터 및 태스크 수준 병렬화 단계에서 할당된 블록내 계산 항목들 각각의 계산을 순차 또는 병렬처리한다. 이때, 상기 CBC-MAC가 병렬처리된다면, 병렬처리를 수행하는 코어들 사이에는 반드시 동기화가 삽입되어 병렬처리된다.

[0152] 정리하면, 상기 멀티코어 프로세서는 상기 처리 대상 태스크인 CTR을 상기 코어 할당 단계에서 할당된 코어들을 통해 병렬처리하고, CBC-MAC을 상기 코어 할당 단계에서 할당된 코어들을 통해 순처 또는 병렬처리한다. 다만, 상기 CTR 병렬 처리를 수행하는 코어들은 상기 데이터 및 태스크 수준 병렬화 단계에서 할당된 데이터 블록들을 병렬 처리하고, 상기 CBC-MAC을 병렬처리하는 경우에는, 블록내 계산 항목들 사이의 종속성을 고려하여 동기화를 삽입하여 병렬처리를 수행하며, 상기 CBC-MAC을 순차처리하는 경우에는 순차적으로 블록내 계산 항목들을 순차처리하면서, 데이터 블록들을 순차 처리한다.

[0153] 한편, 상기 멀티코어 프로세서의 코어 수는 CTR을 데이터 수준에서 병렬처리하고, CBC-MAC을 순차 처리 또는 데이터 수준에서 병렬처리함과 동시에, 상기 CTR 병렬처리와 CBC-MAC 순처 또는 병렬처리를 동시에 병행적으로 처리하기 때문에(태스크 수준에서 병렬처리하기 때문에), 최소 3-코어이면 적용 가능하다. 그러나, CBC-MAC 태스크 처리를 병렬처리하는 것을 포함하고 있기 때문에, 최소 4-코어인 것이 바람직하다.

[0154] 따라서, 상기 멀티코어 프로세서의 코어 수는 4개 이상으로 구성되면, 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리를 수행할 수 있다.

[0155] 상기와 같은 코어 수(4 코어 이상)를 가지는 멀티코어 프로세서의 코어들을 상기 CTR 병렬 처리 및 상기 CBC-MAC 순차 또는 병렬 처리에 할당하는 상기 코어 할당 단계에서는 상기 CTR을 병렬 처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 1/2의 코어들을 할당한다. 즉, 태스크 수준에서 병렬처리를 해야 하기 때문에, 전체 코어의 일부인 절반의 코어들을 할당한다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면 N/2개의 코어를 상기 CTR 병렬 처리를 위하여 할당한다. 이와 같이 전체 코어의 절반을 CTR 태스크 처리에 할당하는 이유는 암호화 부분과 메시지 인증 부분의 계산 시간이 유사하기 때문이다.

[0156] 반면, 상기 코어 할당 단계에서는 상기 CBC-MAC을 순차 또는 병렬처리하기 위하여 상기 멀티코어 프로세서의 코어들 중, 상기 전체 코어들 중에서 상기 CTR 태스크 처리에 할당한 코어를 뺀 나머지 코어들에서 선택하여 할당한다. 구체적으로, CBC-MAC의 태스크 처리가 순차 처리인 경우에는 1개의 코어를 할당하고, 병렬 처리인 경우에는 2개 이상의 코어들을 할당한다.

[0157] 즉, 상기 코어 할당 단계에서는 멀티코어 프로세서의 코어들 중, CRT 태스크 처리에 할당한 코어를 뺀 나머지 코어들 중에서 병렬처리가 가능한 최소 코어 수인 2개 이상의 코어를 상기 CBC-MAC 병렬처리를 위하여 할당한다. 물론, 순차처리인 경우에는 하나의 코어만 할당하면 된다. 예를 들어, 상기 멀티코어 프로세서의 코어 수가 N이면, N/2개의 코어가 상기 CRT 태스크 처리에 할당되었기 때문에, 2개보다 크거나 같고 N/2개보다 작거나 같은 수의 코어를 상기 CBC-MAC 병렬 처리를 위하여 할당할 수도 있다.

[0158] 이와 같이, CBC-MAC에 할당되는 코어 수가 멀티코어 프로세서의 모든 코어에서 CTR 태스크 처리에 할당한 코어 수를 뺀 나머지 수가 아닐 수 있는 이유는 실제 멀티코어 프로세서에서 코어 간 동기화 오버헤드가 상당하고 코어의 개수에 따라 일반적으로 오버헤드가 증가하기 때문이다. 즉, 동기화 오버헤드를 고려하여, 상기 CBC-MAC 병렬 처리를 위한 코어 수를 결정해야 한다.

[0159] 구체적으로 설명하면, 상기 코어 할당 단계에서는 상기 CBC-MAC을 순차 또는 병렬처리 하기 위하여 할당되는 코어 수를 동기화 오버헤드와 CBC-MAC 계산 시간의 상관관계에 의하여 결정한다. 즉, 병렬 처리를 위한 코어 수가 많으면, 동기화 오버헤드가 고려되기 때문에 처리 속도가 항상 빠르지 않는다. 따라서, 동기와 오버헤드와 CBC-MAC 처리 시간의 상관관계를 따져서 최적의 코어 수를 결정해야 한다.

[0160] 이와 같은, 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 순차 또는 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부에 저장하였다가, CBC-MAC 순차 또는 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 순차 또는 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 순차 또는 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0161] 한편, 상기 코어 할당 단계에서는 상기 CBC-MAC을 병렬처리 하기 위하여 상기 블록 내 계산 항목들마다 다른 코

어 수를 할당할 수도 있다. 이 때, 상기 코어 할당 단계에서는 상기 블록내 계산 항목들 각각에 할당되는 코어 수를 각 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 결정한다.

[0162] 즉, 계산 항목들의 처리량이 많다고 하여, 무조건 많은 코어들을 할당하게 되면, 동기화 오버헤드가 커지는 문제점이 발생한다. 따라서, 계산 항목들의 처리량과 동기화 오버헤드를 고려하여 최적의 코어 수를 결정한다. 예를 들어, SubBytes는 1-코어, ShiftRows는 1-코어, MixColumns는 1-코어, AddRoundKey는 2-코어로 할당하는 것과 같이, 각 계산 항목들마다 다른 코어 수를 할당할 수 있다.

[0163] 이와 같은, 각 계산 항목들에 할당될 최적의 코어 수는 사전 실험을 통하여 미리 결정될 수 있다. 따라서 사전에 결정된 최적의 코어 수(CBC-MAC 병렬 처리를 위하여 할당되는 코어 수)를 상기 코어 할당부에 저장하였다가, CBC-MAC 병렬 처리를 수행하고자 하는 경우에, 저장된 코어 수만큼 상기 CBC-MAC 병렬 처리에 할당하면 된다. 물론, 사용자에 의하여 코어 수를 조정할 수 있도록 구성할 수도 있다. 또는, 컴퓨터 시스템 내에서 할당 코어 수를 변경하면서 복수 번의 CBC-MAC 병렬 처리를 수행한 결과를 이용하여, 컴퓨터 시스템 내에서 자동으로 최적의 코어 수를 결정하여 사용할 수도 있다.

[0164] 이상에서 설명한 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬처리 장치를 이용한 병렬 처리 방법에 의하여 CTR 병렬 처리와 CBC-MAC 순차 또는 병렬처리를 동시에 병행적으로 수행하면, 처리 속도가 개선된다. 왜냐하면, CBC-MAC 순차 또는 병렬 처리를 수행할 코어 수를 동기화 오버헤드, CBC-MAC 처리 시간 및 각 계산 항목들의 처리량 등을 고려하여 결정하여, 최적의 코어 수를 활용하기 때문이다.

[0165] 이상에서 설명한 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 방법의 구체적인 예시를 첨부된 도면을 참조하여 살펴본다.

[0166] 도 10 내지 도 12는 본 발명의 제2 실시예인 데이터 수준 병렬화 및 태스크 수준 병렬화를 이용하여 AES-CCM 병렬 처리 방법의 예시를 보여준다. 한편, 도 10 내지 도 12 예시는 4-코어를 이용하여 데이터 수준의 병렬화 및 태스크 수준 병렬화를 이용하여 AES-CCM 병렬 처리를 수행한 예시이다.

[0167] 데이터 수준 병렬화(data-level parallelism) 외에 태스크 수준 병렬화(task-level parallelism)까지 동시에 이용한다면 더 좋은 성능을 기대할 수 있다. 즉, 암호화 부분과 메시지 인증 부분의 계산 시간이 유사하기 때문에, 도 10과 같이 CTR과 CBC-MAC에 각각 2-코어를 할당하여 두 부분을 함께 병렬처리(task-level parallelism)하는 방법을 적용할 수 있다.

[0168] 이 경우에도 데이터 수준 병렬화(data-level parallelism)만을 이용한 본 발명의 실시예 1에서 설명한 예시처럼, 동기화 오버헤드와 CBC-MAC 계산의 상관관계에 의하여 CBC-MAC 부분에 할당될 최적의 코어 수를 도출할 수 있다. 예를 들어, 동기화 오버헤드를 고려할 때 CBC-MAC 부분을 1개의 코어에서 수행하는 것이 2개의 코어를 사용하는 것보다 더 좋은 성능을 제공하는 경우를 도 11에서 나타내고 있다.

[0169] 주의할 점은, 이 경우 블록 데이터간 종속성이 없는 암호화 부분에 3개의 코어를 할당할 수 있으나, 실질적인 성능 개선으로 연결되지는 않는다는 점이다. 즉, 암호화 부분에 3개의 코어를 할당하여 암호화 부분의 수행시간을 1/3로 줄일 수는 있으나, 전체 수행 시간은 암호화 부분과 유사한 계산 처리량을 요하는 메시지 인증 부분에 의하여 결정되기 때문이다.

[0170] 마지막으로, CBC-MAC의 블록내 계산인 SubBytes, ShiftRows, MixColumns, AddRoundKey 각각의 서로 다른 처리량과 동기화 오버헤드를 고려하여 각각의 계산에 최적인 코어를 다르게 설정하는 병렬 처리 방법도 생각할 수 있다. 예를 들어, 도 12에 도시된 바와 같이, SubBytes는 1-코어, ShiftRows는 1-코어, MixColumns는 1-코어, AddRoundKey는 2-코어로 할당될 수도 있다.

[0171] 본 발명에서 제안한 다양한 병렬 처리 방법의 성능을 분석하기 위하여, 먼저 AES 블록내 계산인 SubBytes, ShiftRows, MixColumns, AddRoundKey 각각의 수행 시간을 Rapsim 시뮬레이터[K. Park, et al., "On-Chip Multiprocessor with Simultaneous Multithreading," *ETRI Journal*, Vol. 22, No. 4, pp. 13-24, 2000.]로 측정하였다. 그 결과는 아래 표 1과 같다.

[0172]

AddRoundKey	140
SubBytes	14,707
ShiftRows	26
MixColumns	212

[0173] - 표 1 -

[0174] 그리고, 코어의 개수를 P라 할 때 멀티코어간 동기화를 구현하는 다양한 기법들을 직접 하드웨어로 구현한 방법(cost = O(1)), 효과적인 소프트웨어로 구현한 방법(cost = O(log*P*)), 효과적이지 못한 소프트웨어로 구현한 방법(cost = O(*P*))으로 구분할 수 있다.

[0175] 예를 들어, 도 13과 같은 동기화 오버헤드를 갖는 가상의 4-코어 프로세서 A, B, C가 있다고 가정하자. 그러면, 각각의 프로세서를 이용하여 AES 블록내 계산인 SubBytes, ShiftRows, MixColumns, AddRoundKey를 1-코어, 2-코어, 3-코어, 4-코어로 수행한 결과를 요약하면 아래 표 2와 같다.

[0176]

	1-core	2-core	3-core	4-core
A	150,779	87,989	62,859	50,294
B	150,779	92,189	76,887	71,294
C	150,779	96,389	81,759	79,694

[0177] - 표 2 -

[0178] 또한, 상기 표 2를 이용하여 100 블록 크기의 데이터를 AES-CCM으로 처리하는 경우, 본 발명에서 제안한 병렬처리 방법들의 수행 시간을 계산하면 아래 표 3과 같다.(계산의 편의상, KeyExpansion 시간은 포함하지 않았다.) 즉, 프로세서 A에서는 방법1-1과 방법2-1, 프로세서 B와 프로세서 C에서는 방법2-1이 가장 좋은 성능을 제공할 수 있다.

[0179]

프로세서	방법1-1	방법1-2	방법1-3	방법2-1	방법2-2	방법2-3
A	8,798,950	12,568,425	12,620,150	8,798,950	15,077,900	15,630,700
B	10,898,950	12,988,425	14,022,887	9,218,950	15,077,900	15,840,700
C	11,738,950	13,408,425	290,203	192,779	301,558	321,014

[0180] - 표 3 -

[0181] 여기서 방법 1-1은 CBC-MAC 전체 4-Core, 방법 1-2는 CBC-MAC 전체 2-Core,방법 1-3은 CBC-MAC 중, SubBytes : 2-Core, ShiftRows : 1-Core, Mixcolumns : 2-Core, AddRoundKey : 3-Core, 방법 2-1은 CBC-MAC 전체 2-Core, 방법 2-2는 CBC-MAC 전체 1-Core, 방법 2-3은 CBC-MAC 중, SubBytes, ShiftRows, Mixcolumns : 1-Core, AddRoundKey : 2-Core로 설정함

[0182] 이상은 본 발명에서 제안한 다양한 병렬 처리 방법의 성능을 비교하기 위하여 가상의 4-코어 프로세서를 가정하였으나, 실제 4-코어 프로세서를 이용하여 100 블록 크기의 데이터를 AES-CCM으로 처리한 결과는 다음과 같다.

[0183] 먼저 다양한 프로세서 환경을 가정하여 Pthread[B. Barney, POSIX Threads Programming, http://www.llnl.gov/computing/tutorials/pthreads, 2006.]로 부하 분산 방법을 구현하였다. 그리고, Intel Core2 Quad CPU(2.33 GHz)와 AMD Phenom II CPU(3.2 GHz)로 측정한 동기화 오버헤드를 아래 표 4에 정리하였다.

[0184]

	2- core	3- core	4- core
Intel	8,732	13,130	16,854
AMD	73,219	116,328	206,025

[0185] - 표 4 -

[0186] 예상과 달리 두 개의 프로세서 모두에서 상당한 크기의 동기화 오버헤드가 있음을 확인하였는데, 이는 Pthread 프로그램을 수행하기 위해 포팅한 Linux가 실제 프로세서들의 특성을 최대한 이용하지 못한 것으로 판단된다. 그리고, 이렇게 높은 동기화 오버헤드를 고려하면, 상기 방법 2-2가 Intel(759,500 cycles)과 AMD(2,420,000 cycles) CPU 모두에서 가장 좋은 성능을 제공함을 확인하였다. 참고로 Intel과 AMD CPU의 1-코어를 이용한 순차처리는 각각 1,519,000 cycles과 4,840,000 cycles가 소요되었다.

도면의 간단한 설명

[0187] 도 1은 본 발명에 적용되는 AES-CCM의 처리 과정에 대한 순서도이다.

[0188] 도 2는 본 발명에 적용되는 CTR 운용모드의 처리 과정을 설명하기 위한 이해도이다.

[0189] 도 3은 본 발명에 적용되는 CBC-MAC 운용모드의 처리 과정을 설명하기 위한 이해도이다.

[0190] 도 4는 본 발명에 적용되는 AES-CCM 운용모드의 전체 처리 과정을 설명하기 위한 이해도이다.

[0191] 도 5는 본 발명의 제1 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치의 구성도이다.

[0192] 도 6은 본 발명의 제2 실시예에 따른 멀티코어 프로세서를 이용한 AES-CCM 병렬 처리 장치의 구성도이다.

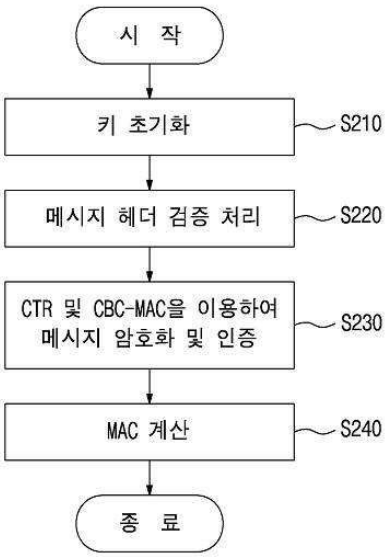
[0193] 도 7 내지 도 9는 본 발명의 제1 실시예인 데이터 수준 병렬화을 이용하여 AES-CCM 병렬 처리 방법의 예시를 보여주기 위한 예시도이다.

[0194] 도 10 내지 도 12는 본 발명의 제2 실시예인 데이터 수준 병렬화 및 태스크 수준 병렬화를 이용하여 AES-CCM 병렬 처리 방법의 예시를 보여주기 위한 예시도이다.

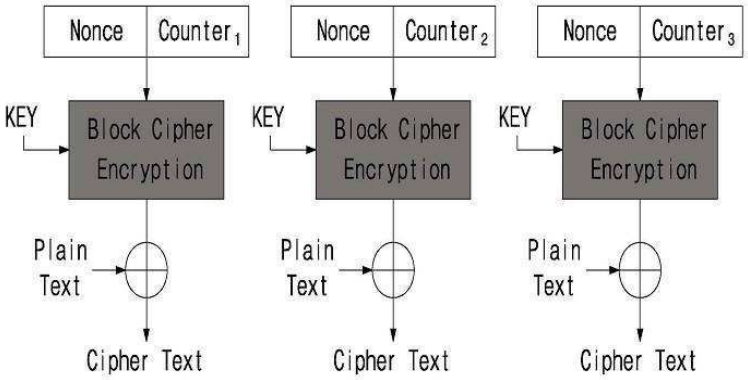
[0195] 도 13은 본 발명의 성능 검증을 위해 사용된 동기화 오버헤드를 갖는 가상의 4-코어 프로세서를 보여주는 그래프이다.

도면

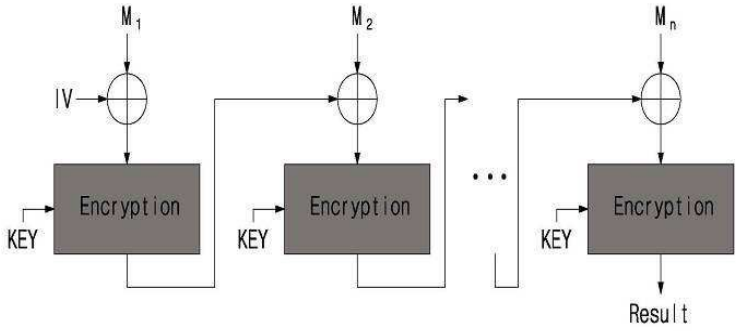
도면1



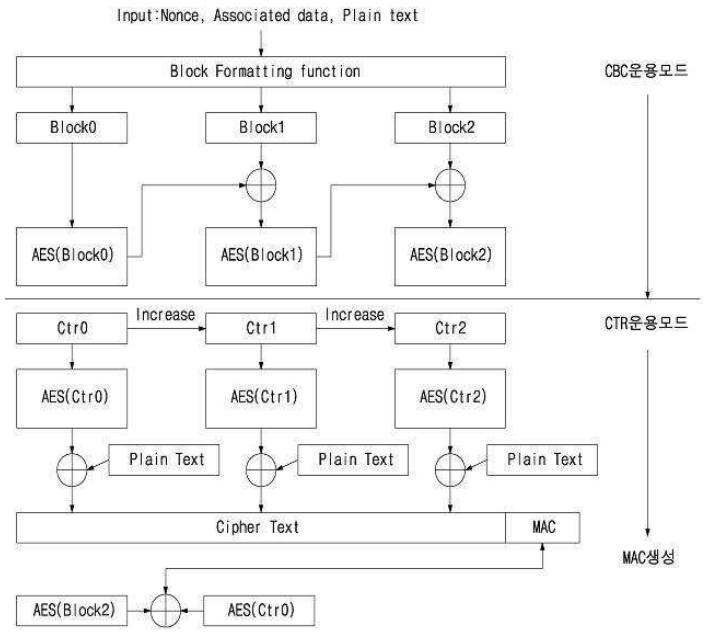
도면2


Nonce
Counter1
Nonce
Counter2
Nonce
Counter3
KEY
Block Cipher Encryption
KEY
Block Cipher Encryption
KEY
Block Cipher Encryption
Plain Text
Plain Text
Plain Text
Cipher Text
Cipher Text
Cipher Text

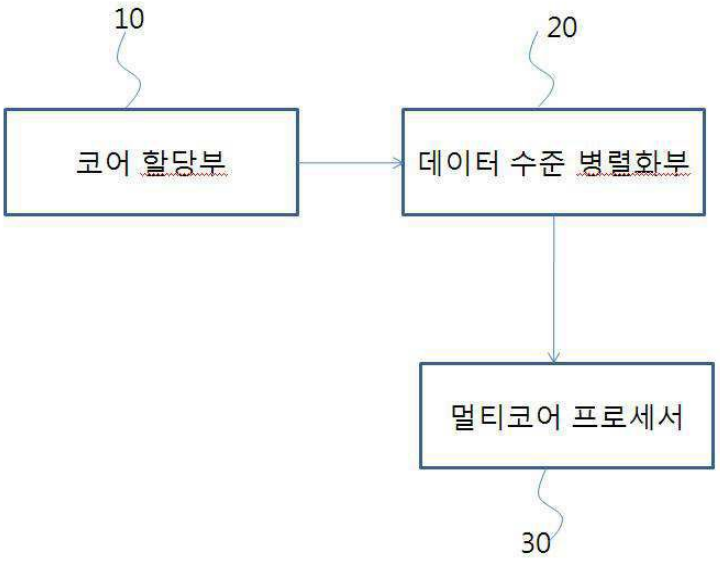
도면3


M1
M2
Mn
IV
Encryption
Encryption
Encryption
KEY
KEY
KEY
Result

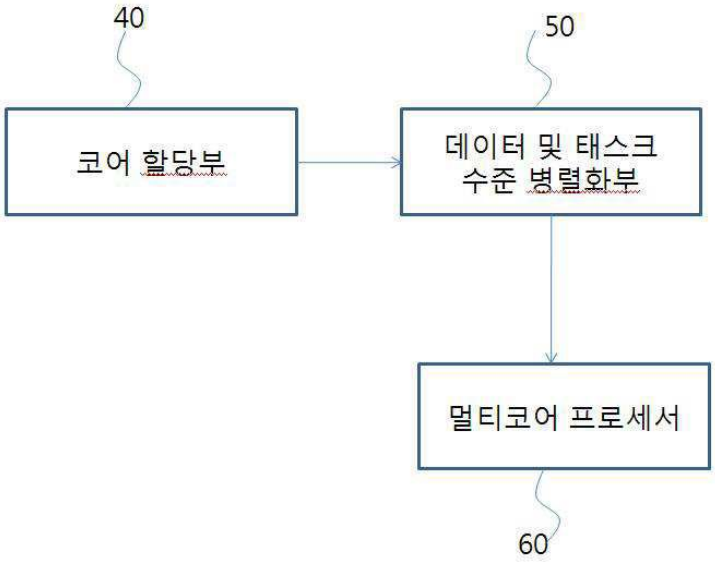
도면4


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)

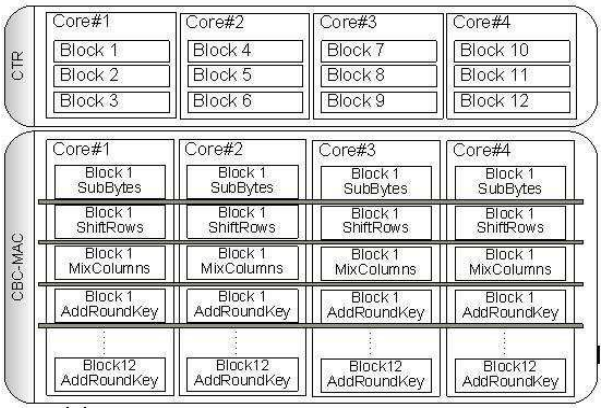
도면5


10
20
코어 할당부
데이터 수준 병렬화부
멀티코어 프로세서
30

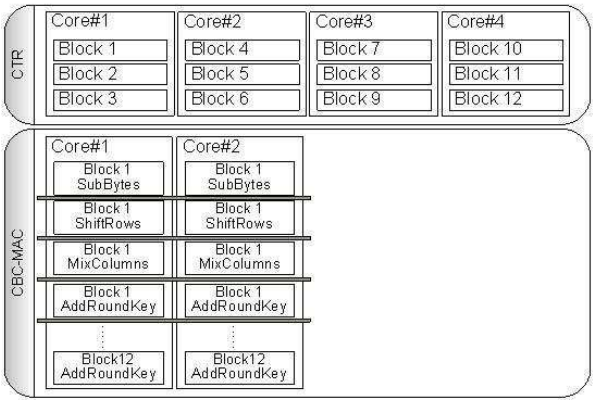
도면6


40
50
코어 할당부
데이터 및 태스크
수준 병렬화부
멀티코어 프로세서
60

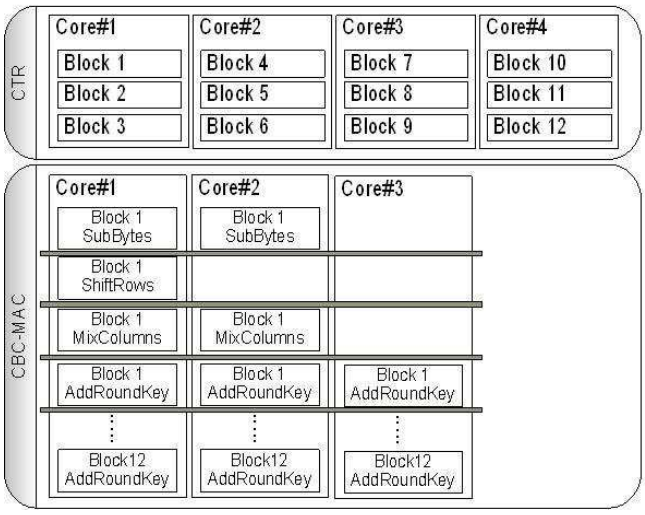
도면7


CTR
Core#1
Block 1
Block 2
Block 3
Core#2
Block 4
Block 5
Block 6
Core#3
Block 7
Block 8
Block 9
Core#4
Block 10
Block 11
Block 12
CBC-MAC
Core#1
Core#2
Core#3
Core#4
Block 1 SubBytes
Block 1 ShiftRows
Block 1 MixColumns
Block 1 AddRoundKey
Block12 AddRoundKey

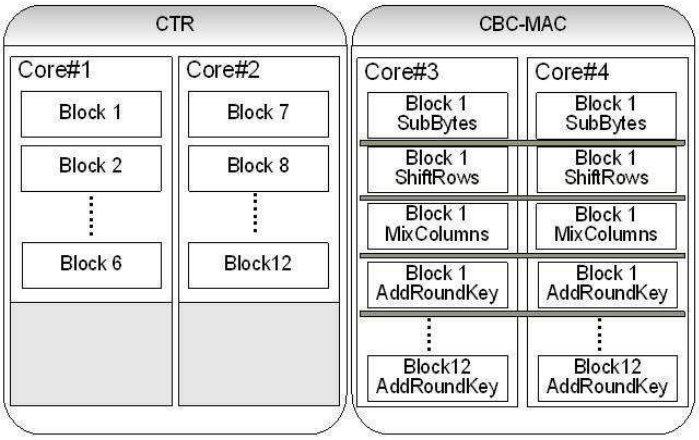
도면8


CTR
Core#1
Block 1
Block 2
Block 3
Core#2
Block 4
Block 5
Block 6
Core#3
Block 7
Block 8
Block 9
Core#4
Block 10
Block 11
Block 12
CBC-MAC
Core#1
Core#2
Block 1 SubBytes
Block 1 ShiftRows
Block 1 MixColumns
Block 1 AddRoundKey
Block12 AddRoundKey

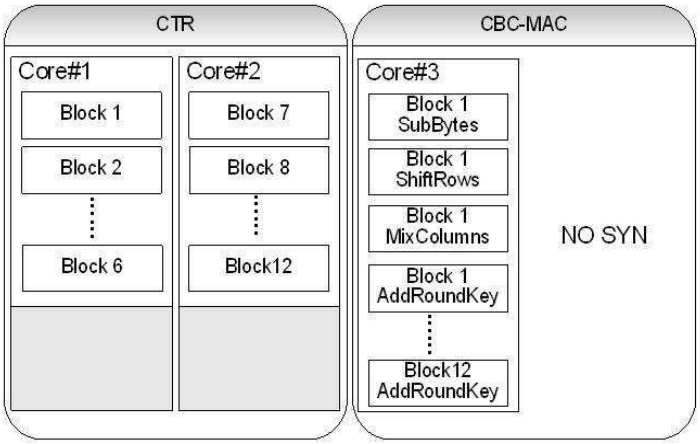
도면9


CTR
Core#1
Block 1
Block 2
Block 3
Core#2
Block 4
Block 5
Block 6
Core#3
Block 7
Block 8
Block 9
Core#4
Block 10
Block 11
Block 12
CBC-MAC
Core#1
Core#2
Core#3
Block 1 SubBytes
Block 1 ShiftRows
Block 1 MixColumns
Block 1 AddRoundKey
Block12 AddRoundKey

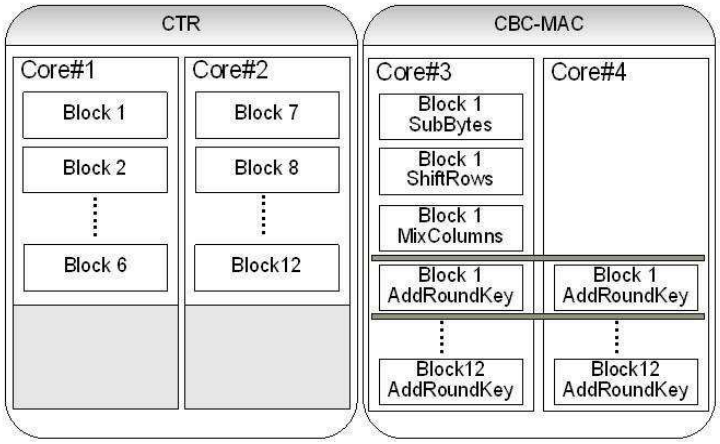
도면10



도면11



도면12



도면13

