2003년도 대한전자공학회 하계종합학술대회 논문집 제26권 제1호

IPSec Engine의 외부 인터페이스 확장을 위한 범용 메모리 컨트롤러의 설계

김철민, 임용준, 김영근
삼성전자 디지털미디어연구소
모바일플랫폼 연구실

Design of Novel Memory Controller to Extend IPSec External Interface

ChulMin Kim, YongJun Lim, YoungKeun Kim
Mobile Platform Lab. DM R&D Center
Samsung Electronics Co., Ltd.
E-mail : chulmin73.kim@samsung.com

Abstract

본 논문은 기존 IPSec(internet protocol security) 하드웨어의 고정된 인터페이스 방식을 개선한 동적 인터페이스 방식의 IPSec 을 구현하기 위하여 범용 인터페이스 메모리 컨트롤러를 제안하고 있다. 범용 컨트롤러는 다양한 플랫폼 상의 외부 인터페이스와 내부 암호화 모듈 간 데이터 폭 차이를 상쇄하여 다양한 환경 하에서 구동이 가능한 하드웨어 인터페이스를 제공할 수 있을 것이다.

I. 서론

최근 차세대 인터넷 보안의 핵심 기술인 IPSec 에 사용되는 다양한 암호화 알고리즘을 하드웨어로 구현하여 리소스 및 연산 시간을 획기적으로 개선하기 위한 연구가 진행되고 있다. [1] 하드웨어 구현은 정적인 특성 때문에 내부 동작은 물론, 외부 인터페이스 설계가 매우 중요한데, 기존의 IPSec 엔진은 입출력 인터페이스가 고정되어 있으므로, 메모리 컨트롤러를 통해 저장된 패킷 데이터를 IPSec 엔진의 고정된 데이터 폭에 맞추어 입력하는 방식을 사용한다. 그러나, 외부 인터페이스가 변경되거나 암호화 알고리즘을 새로 확장하는 경우 기존에 설계된 하드웨어는 재사용할 수 없으며, 메모리 컨트롤러를 포함한 IPSec 전체를 다시 설계 해야 하는 불편함이 발생한다. 그러므로, 본 논문에서는 인터페이스 변경 또는 알고리즘 확장 시에도 하드웨어 자체의 설계 변경 없이 연결 방식(mode)의 선택만으로 직접 적용 가능한 범용 메모리 컨트롤러를 제안하고 있다. 범용 메모리 컨트롤러 방식은 인터넷 보안기술의 규약(protocol)에 따라 채용될 수 있는 다양한 암호화 및 복호화 알고리즘과 직접 연동 가능한 가변 폭(flexible width) 메모리 컨트롤러 및 가변적인 플랫폼 상에서 외부 인터페이스를 연동하기 위한 패킷(packet) 데이터 변환 장치로 구성되며, 장치의 동작 모드를 환경에 따라 설정 가능하도록 설계되어 있다. 제안하는 방식을 통해 차세대 인터넷의 다양한 플랫폼과 알고리즘 상에서 구동 가능한 하드웨어 IPSec 을 설계할 수 있을 것이다

II. 기본 동작

IPSec 에서 사용하는 암호화 모듈은 암호화 정책에 따라 개선된 알고리즘으로 항상 확장이 가능해야 하며, 다양한 플랫폼 상에서 구동될 수 있어야 한다. 이때, 각각의 암호화 알고리즘은 고정된 고유의 입출력 데이터 폭이 있는 반면, IPSec 을 사용하는 환경은 다양한 인터페이스를 요구할 수 있다. [1] [2]

기존의 방식에서는 IPSec 을 하드웨어로 구현하기 위하여 개별 알고리즘마다 외부의 정적 메모리컨트롤러를 설계하여 각각의 외부 모듈로부터 입력 받은 패킷

데이터의 폭을 조절하여 암호화 모듈에 제공하는 방식을 사용하고 있다. 그러나, IPSec 외부의 인터페이스는 사용되는 환경에 따라 얼마든지 가변적일 수 있으며, 내부의 알고리즘 또한 추가 및 변경이 가능하므로, 확장이 필요한 경우 외부 인터페이스와 알고리즘 데이터 폭에 맞도록 메모리 컨트롤러 자체를 다시 설계해야 하는 불편함이 따른다.

그러므로, 본 논문에서는 외부 인터페이스 모듈로부터 입력되는 임의의 N bits 데이터 스트림(stream)을 내부 m bits 메모리의 데이터 폭에 맞도록 조절하는 가변 인터페이스 구성 방법 및 내부 m bits 메모리와 IPSec 엔진의 k bits 간의 가변 인터페이스 및 IPSec 엔진의 상태에 따라 메모리 액세스를 컨트롤 하는 범용 메모리 컨트롤러를 제안하여 비효율적인 기존의 방식을 개선하고자 한다. (그림 1.1) 은 제안하는 방식의 알고리즘을 보여 주고 있다.

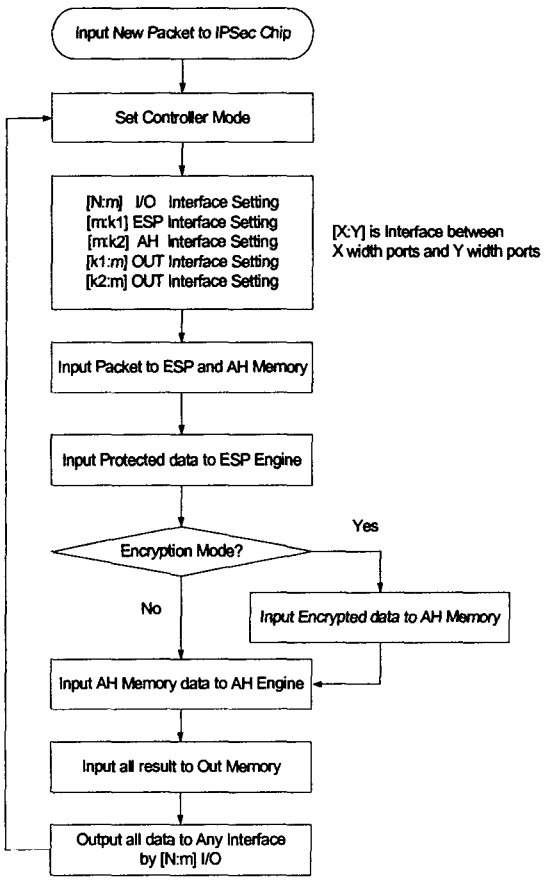

그림 1.1 전체 알고리즘

III. 하드웨어 세부 구조

가변적인 내부 암호화 알고리즘과 외부 환경 하에서 적용 가능한 동적 인터페이스를 구현하기 위한 범용 메모리 컨트롤러는 크게 동적 메모리 컨트롤러 및 패킷 데이터 변환 부로 구성되어 있다.

동적 메모리 컨트롤러는 외부로부터 입력되는 N bits 데이터 스트림의 폭을 내부 m bits 메모리 폭으로 변환하여 저장하는 역할을 한다. 즉, 암호화 과정에서 처리되는 고정 폭의 데이터 형식을 생성하기 위해 임시로 저장되는 고정된 메모리 공간상의 데이터 폭이 임의의 외부 인터페이스 모듈로부터 입력되는 가변 폭 데이터의 배수인 경우, 고정 폭을 가변 폭으로 나눈 값을 결합 값의 개수만큼의 가변 폭의 데이터 스트림(stream)을 차례로 입력 받게 된다.

순서대로 입력되어 메모리 상에 저장된 데이터 패킷은 IPSec 엔진의 요구에 따라 데이터 패킷 변환 부를 통해 다시 암호화 알고리즘에 적절한 데이터 폭으로 결합된 새로운 데이터 패킷을 생성하여 출력되게 된다. 이때, 데이터 스트림의 암호화 과정에서 처리되는 데이터 패킷의 폭인 고정 폭이, 임의의 암호화 알고리즘에서 사용되는 데이터 패킷의 폭인 가변 폭의 배수인 경우 고정 폭을 가변 폭으로 나눈 값인 결합 값 개수의 가변 폭의 데이터 패킷을 차례로 출력하게 된다. 다음 순서대로 입력된 결합 개수의 가변 폭의 데이터 패킷을 결합하여, 고정 폭의 데이터 패킷을 생성하여 출력하는 단계를 거쳐 출력된 고정 폭의 데이터 패킷을 암호화한 후, 고정 폭의 암호화 데이터 패킷을 생성하여 출력하게 된다.

가변 폭 암호화 데이터 패킷의 복호화 방법은 복호화 과정에서 처리되는 암호화 데이터 패킷의 폭인 고정 폭이, 임의의 인터페이스 모듈로부터 입력되는 임의의 암호화 데이터 패킷의 폭인 가변 폭의 배수인 경우, 고정 폭을 가변 폭으로 나눈 값인 결합 값의 개수의 가변 폭의 암호화 데이터 패킷을 차례로 입력 받고, 차례로 입력된 결합 값의 개수의 가변 폭의 암호화 데이터 패킷을 결합하여, 고정 폭의 암호화 데이터 패킷을 생성하여 출력하는 단계 및 출력된 고정 폭의 암호화 데이터 패킷을 복호화하여, 고정 폭의 데이터 패킷을 생성하여 출력하는 단계로 구성된다. [3] [4]

그러므로, 제시한 방식을 통하여 IPSec 하드웨어에 연결되는 외부 인터페이스 모듈이 변경된 경우, 임의의

외부 인터페이스 모듈로부터 출력되는 가변 폭의 데이터 패킷을 암호화하고, 가변 폭의 암호화 데이터 패킷을 복호화할 수 있다. (그림 1.2)는 앞서 설명한 동적 메모리 컨트롤러가 가변 폭의 외부 데이터 입력으로부터 고정 폭의 내부 메모리에 데이터를 저장하게 하기 위한 하드웨어 구조를 보여주고 있다.

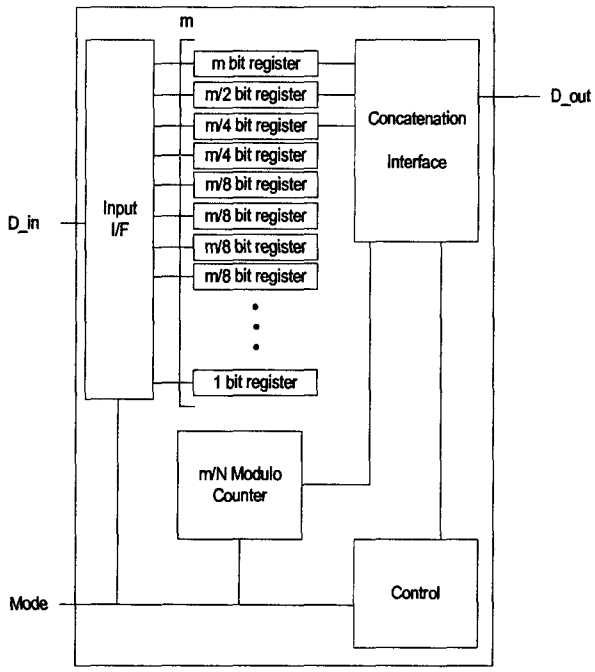

그림 1.2 동적 메모리 컨트롤러

(그림 1.2)에서 D_in 은 N bits 의 임의의 인터페이스에 연결 될 수 있는 가변 입력 단자(ports)이며, 입력 데이터 폭은 Mode 단자를 통하여 설정이 가능하다. 입력된 데이터는 입력 인터페이스 모듈을 통해 내부 메모리의 데이터 폭에 따라 1 bit 에서 m bit 폭의 레지스터(register)에 저장되어 내부 메모리에 순서대로 저장되게 된다.

(그림 1.3)은 본 논문에서 제안하고 있는 전체 하드웨어 아키텍처를 보여주고 있다. (그림 1.3)에서 외부 입력된 임의 폭의 데이터는 (그림 1.2)의 동적 메모리 컨트롤러를 통해 해당 알고리즘의 내부 메모리에 고정 폭 데이터로 저장된 후 IPSec 엔진의 요구에 따라 해당 알고리즘의 입력에 맞는 데이터 형태로 변환되어 출력된다. 출력된 데이터는 각각의 알고리즘을 통해 암호화 및 복호화 된 후 다시 동적 메모리 컨트롤러를 통해 임의 폭의 외부 인터페이스로 변환되어 외부로 출력된다. [5]

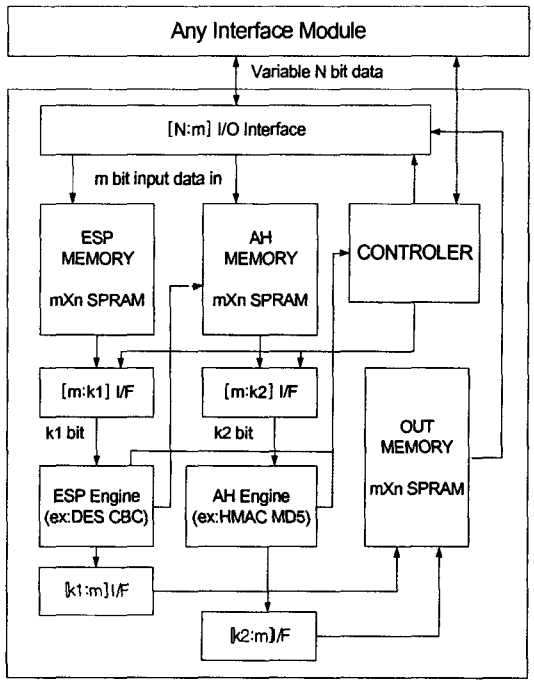

그림 1.3 전체 하드웨어 구조

IV. 결론

제안하는 방식은 외부에서 메모리 컨트롤러의 동작 모드를 선택하여, 가변 폭의 패킷 데이터 입력으로부터 가변 폭의 패킷 데이터 출력을 생성할 수 있는 구조를 가지기 때문에 외부 인터페이스의 변경이나, 내부 알고리즘의 추가 시에도 설계 변경 없이 동작 조건을 설정하는 간단한 조작만으로 IPSec 엔진을 구동시킬 수 있다. 그러므로, 본 설계를 통해 임의의 플랫폼 상에서도 추가 확장이 용이한 IPSec 제작이 가능하다.

본 발명에 따르면, IPSec 칩에 연결되는 외부 인터페이스 모듈이 변경된 경우, 임의의 외부 인터페이스 모듈로부터 출력되는 가변 폭의 데이터 패킷을 암호화하

고, 가변 폭의 암호화 데이터 패킷을 복호화할 수 있다는 효과가 있다. 특히, IPv6 환경에서는 IPv4 환경과는 달리, 특성상 다양한 플랫폼(예를 들면, 가정내의 모든 가전기기)에서 구동되어야 할 것이므로, 외부 인터페이스 모듈의 변경이 빈번하게 발생할 것이고, 이것으로 말미암아, IPSec 칩을 변경하여 재설계하는 경우에도, 메모리 컨트롤러 상위 단의 인터페이스 부분만을 재설계하면 되기 때문에, 메모리 컨트롤러를 포함한 IPSec 코어를 재설계할 필요가 없다는 효과가 있다.

참고문헌

[1] S.Deering and R.Hinden, "Internet Protocol, Version6(IPv6) Specification", IETF RFC 2460, December 1998.

[2]C. Madson and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm", ", IETF RFC 2405, November 1998

[3] P. Karn and P. Metzger, "The ESP DES-CBC Transform", IETF RFC 1829, August 1995.

[4] S. Kent and R. AtKinson, "Security Architecture for Internet Protocol", 1 IETF RFC2401, November 1998.

[5] S. Kent and R. AtKinson, "IP Encapsulating Security Payload", 1 IETF RFC2406, November 1998.