RE-CONFIGURABLE PROGRAMMABLE SECURITY PROCESSOR FOR NETWORK APPLICATIONS

Zaheer Ahmed, M. Mohsin Rahmatullah, Habibullah Jamal

University of Engineering and Technology,
Taxila, Pakistan

zaheer@carepvtltd.com, (+92-51-2874794), 19-Ataturk Avenue, G-5/1, Islamabad Pakistan
mohsin@carepvtltd.com, (+92-51-2874794), 19-Ataturk Avenue, G-5/1, Islamabad Pakistan
drhjamal@uettaxila.edu.pk, (+92-596-9314224), UET Taxila, Pakistan

ABSTRACT

One of the fundamental challenges for the designers of the crypto processors is to provide the wire-speed security for network applications. In today's ultra-competitive marketplace, the low cost and flexibility for newer algorithms are the key requirements for any crypto processor. The traditional architectures employ hard coded state machines to achieve high data rates, which are useless for new algorithms. This paper describes a Re-configurable and programmable Security Processor with powerful instruction sets to cater for current as well as future security algorithms. The architecture is specially designed and optimized for IPSec applications, involving authentication, encryption, key generation and digital signature generation/verification. The novel architecture can be used with any network processor for wire speed security and can be configured for any data rates by configuring the number of crypto engines. The programmable and configurable architecture increases the life of the processor by incorporating support for future algorithms.

KEY WORDS

IP Security (IPSec), Security Processor, Re-configurable, Authentication, Cipher, Key Generation

RE-CONFIGURABLE PROGRAMMABLE SECURITY PROCESSOR FOR NETWORK APPLICATIONS

1 INTRODUCTION

The processor architectures are categorized as general-purpose processors having a very high degree of flexibility and high power consumption and application specific processors that have low degree of programmability and low power consumption [1]. However, the general purpose processors suffer from very low performance that hinder them from being suitable for many applications, such as wire speed cryptography and Gigabit routers. On the other hand, ASICs (Application Specific Integrated Circuits) are optimized to perform specific operations to attain higher performance. Application Specific processors are not suitable for applications that require high degree of programmability. Moreover ASIC based solution carries with it a number of disadvantages. The development cycle for ASICs tends to be significant, in terms of both time and cost. The time it takes to develop an ASIC, introduces the problem that a solution may be ready to be deployed in the market only after the solution has become obsolete.

Domain Specific Processors are in between General Purpose and Application Specific processors. They have low programmability for general-purpose applications but high programmability, high performance and low power for particular application domain. They are ideal for many applications if their degree of programmability is enhanced while maintaining the same high performance. The Security Processor for IPSEC applications best fits in the domain specific processors category. The cryptography is ever been a hot topic of research, new security algorithms keep on emerging due to advances in cryptography and computing power. Existing algorithms become vulnerable to new attacks also paving the way for new algorithms. The hardwired solutions become obsolete as a consequence and require redesign of the chip. In this paper we present a domain specific programmable security processor that caters for the evolving needs of IPSEC application and provides a platform to support a wide range of algorithms.

This paper is divided into seven sections. Section II illustrates the Top Level architecture of security processor. Section III describes the Authentication, Cipher and Key generation Engines' architecture, section IV explains the InterConnect Engine functionality, section V gives the details of the Interfaces of the security processor. Section VI & VII discusses the results and conclusion.

2 TOP LEVEL ARCHITECTURE

The configurable & programmable Security Processor architecture provides scalable and modular hardware-software solution. The architecture is optimized to process all algorithms associated with IPSEC and IKE. The security processor acts as a slave device to the host, off-loading the host from the data-intensive cryptographic operations used in IPSEC and IKE.

Security processor has programmable cryptographic and public key engines thus allowing the customer to protect its investments in silicon. New encryption / hashing / public-key algorithms can be readily ported in the architecture achieving rapid time to market. The processor can be configured for different authentication (MD5, SHA-1, SHA256, SHAH384, SHA512, RIPEMD-128, RIPEMD-192, TIGER) and encryption (DES, 3DES, AES or any proprietary) algorithms. In our architecture powerful instruction sets are provided to cater for current as well as future authentication and encryption algorithms. The security processor includes an optimized Key Generation engine that can be used for key generation and digital signature generation and verification. A specially designed interconnect engine, designed to schedule the tasks for different cryptographic engines, is the key element to achieve high data rates, involving minimum host interactions.

Figure 1 depicts the Top Level diagram of the security processor architecture. The architecture can be extended by employing multiple layers of cryptographic engines. The CipherHash block & Key Engine block represents one layer of the architecture. The Interconnect & DMA engine has the capability to serve multiple layers of security processor. The multi-layered novel architecture can be used for wire speed security and can be configured for any data rates by configuring the number of layers of security processors and the number of engines in each layer. Each layer & engine can be put in sleep mode during idle time to save the power. The processor can be used with any network processor for IPSEC implementation for high data rates. The programmable engines are optimized for current algorithms and powerful functions are incorporated to support future algorithms.

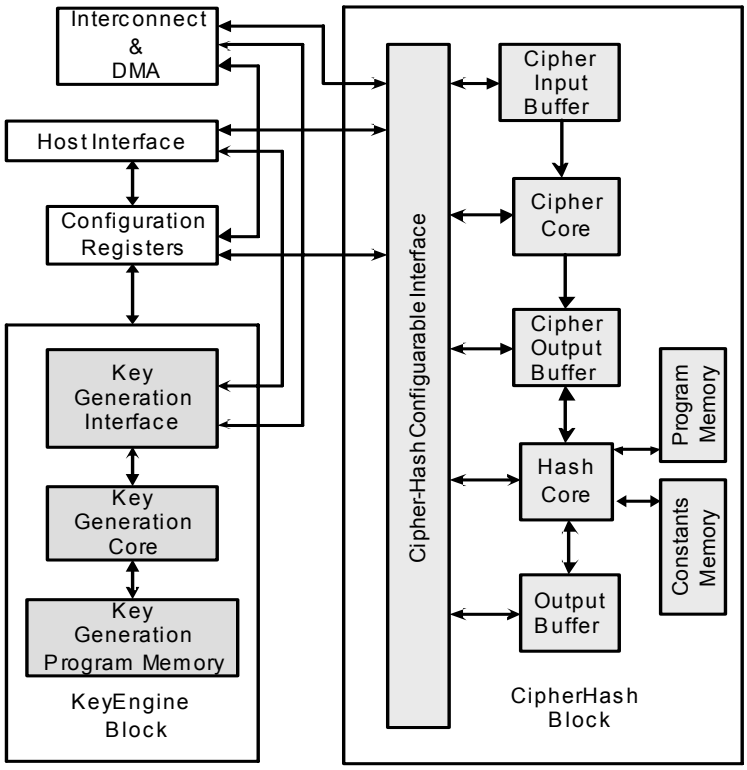


Figure 1: Security Processor TopLevel Architecture

3 CRYPTO ENGINES

Security Processor comprises of authentication, cipher & key generation engines for performing computationally intensive functions.

3.1 Authentication Engine

The Authentication Engine is optimally designed for SHA-1/256/384/512, MD5 [2] and their HMAC [3] versions. The algorithms like RIPEMD-128/192 and TIGER can be easily ported on the same hardware. The architecture of the authentication engine is shown in Figure 2. The engine has the following features.

- Independent data paths for Message Generation Unit (MGU) and Message Computation Unit (MCU).
- Multi Operand Single cycle 64-bit addition support.
- 64-bit Multi-Operand-Complex-Operation Logic Unit (Function Generator).
- 64-bit Sigma Generation Unit for simultaneously rotation/shifting operation on single/multiple operand.
- Two Data Register files, 16x64-bit wide.

- 8Kbytes data memory with word, dual word and long word access.
- 512 x 64 program memory with 64-bit wide instruction word.
- Two level of nesting for zero overhead loops.
- Four level of nesting for subroutine calls.
- Conditional/Unconditional Jumps support.

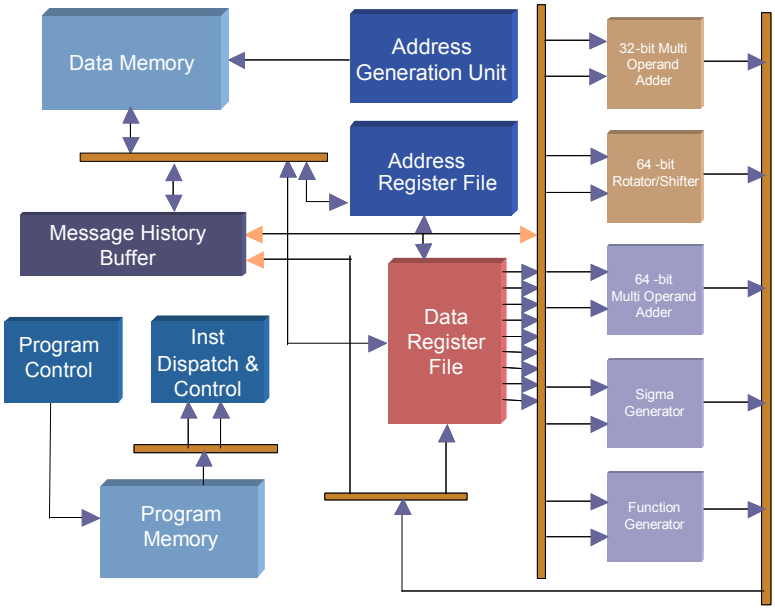


Figure 2: Authentication Engine

The authentication engine is provided with different configuration registers to configure different processing elements inside the engine. The configurations of different processing elements enable the same functional blocks to be utilized for different algorithms. Some of the targeted configurations are listed below to illustrate the power of the engine for new algorithms.

- **General Configuration:**

The programmer can perform following configurations:

- Algorithm type 32 /64 bit
- Number of registers of data register file to be shifted.
- Normal Load
- Load with Ipad (Inner Pad)
- Load with Opad (Outer Pad)
- Load with Comparison, used in comparison of MSG Digest

- **Pad Configuration**

Register is used for configuring data padding. Three bits points to the byte at which to append padding sequence & seven bits for Padding Byte.

- **Message Computation Unit (MCU) Configuration**

A set of 18-bit wide registers. They are used to configure rotation indexes for MCU sigma generator.

- **Message Generation Unit (MGU) Configuration**

A set of 18-bit wide registers, used to configure rotation indexes for MGU sigma generator.

- **Function generator Configuration:**

These are four 18-bit wide registers. They are used to configure Boolean equation to perform different Boolean operations. The programmer can configure different sets of the Boolean functions, based on the configuration bits. The function generator provides all the combinations of

Boolean operations in 2-operand of function XY, YZ. The type of relation between XY and YZ can be selected by configuring the function set bits 17:15 as indicated in the table 1. The table lists the methodology of configuring different function groups Fn.

Function	Bit No	Name	Description
XY (= (!)X (&/|/^) (!)Y)	0	Xbar	X is inverted then anded, ored, exored with operand Y.
	1	Ybar	operand Y is inverted then anded, ored, exored with operand Y.
	3:2	X(?) Y	Specifies the function to be perform on the two operands of function XY. 00 => function output is X. 01 => function output is X & Y. 10 => function output is X | Y. 11 => function output is X ^ Y.
XYYZ (= XY (&/|/^) YZ)	14:12	XY (?) YZ	Specifies the function to perform on the two operands of function XYYZ. 000 => output is XY. 001 => output is XY & YZ. 010 => output is XY | YZ. 011 => output is XY ^ YZ. 100 => function output is YZ.
Fn (=XYYZ (&/|/^) ZX)	17:15	XYYZ (?) ZX	Specifies the function to be performed on the two operands of function Fn. 000 => output is XYYZ. 001 => output is XYYZ & ZX. 010 => output is XYYZ | ZX. 011 => output is XYYZ ^ ZX. 100 => function output is ZX.

Table 1: Function Generator Configuration Table

3.2 Cipher Engine

The Cipher Engine is optimized for DES/TDES/AES [4][5] based encryption/decryption algorithms and provides CBC/ECB/OFB/CFB modes for encryption/decryption. Additional, proprietary (or future) algorithms can also be ported on this engine using the instruction set provided. The block diagram of the cipher engine is depicted in Figure 3; the engine has the following features:

- Support for 128/192/256 bit keys for AES algorithms.
- 110K NAND gates equivalent design.
- Throughput of upto 1.2Gbps
- Programmable and configurable controller optimized for DES/AES algorithms that minimize the overhead.

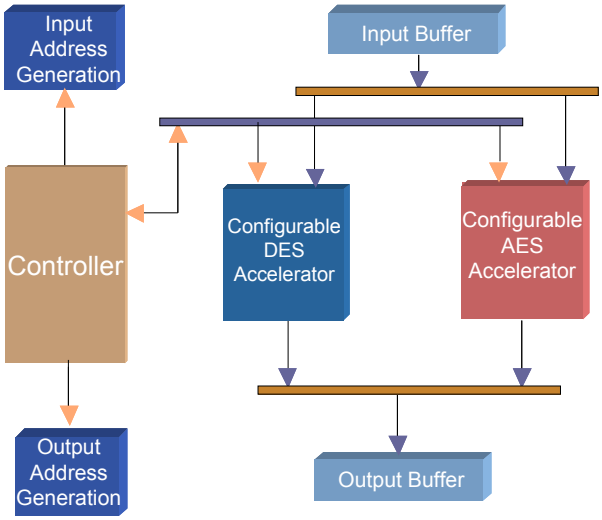


Figure 3: Cipher Engine

3.3 Key Generation Engine

The engine is designed to perform key generation for Diffie-Hellman key exchange, Pseudo-Random Number Generation and Digital Signature Generation and verification (DSA). The base of all these algorithms is following modulo arithmetic's [6][7]:

- Montgomery Modular Reduction
- Montgomery Modular Addition/Subtraction
- Montgomery Modular Exponentiation (used for squaring and inverse also)
- Montgomery Modular multiplication

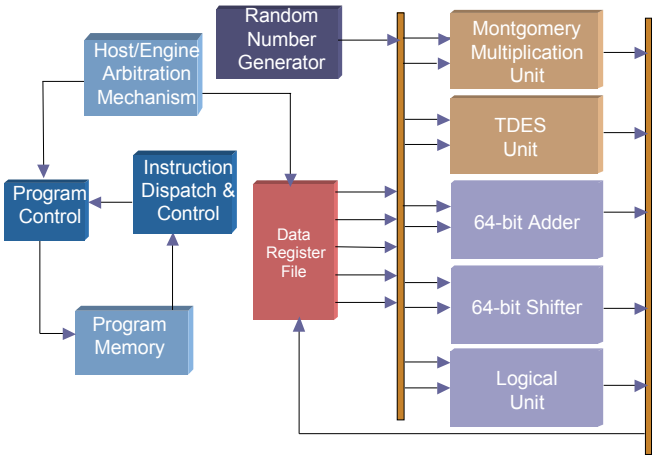


Figure 4: Key Generation Engine

The engine is depicted in Figure 4 and has the following key features:

- 158K NAND gates equivalent design
- Support for upto 512 x 512 bit multiplication
- 512 x 32 bit program memory and 32-bit instruction bus
- 2 levels of nesting for zero overhead loop
- Hardware support for four levels of nesting for subroutine calls
- Conditional and unconditional jumps
- Montgomery Modulo Multiplication unit with support for 160, 512 and 1024 bit operand
- 64-bit adder with support for multiprecision addition, subtraction and comparison

- 64-bit barrel shifter with support for multiprecision left and right shifts
- 16-bit logical unit
- Triple DES unit for ANSI X9.17 Pseudo Random Number Generation
- Pulse Counter register for optional random seed generation

4 INTERCONNECT ENGINE

The InterConnect engine provides the host access to the security processor. It contains a task queue for the host where host can schedule tasks for different Engines. The task can be for any one of the engines or scheduled to pass through all of the three engines. The InterConnect engine decodes the task, performs the data routing on different engine and maintains the status of scheduled tasks. It is also responsible for arbitrating all the data and program code transfer requests for different engine and filling the proper DMA channels with appropriate address and data size information. When any engine completes a particular task, the InterConnect performs the data transfer to either external memory or to any other engine based on the instruction decoded. After completing the task InterConnect sets the task complete flag, which could be used to generate an interrupt for the host, and proceed to perform the next task in the queue. As shown in Figure 5, the InterConnect architecture is flexible enough to cater for multiple layers of crypto engines.

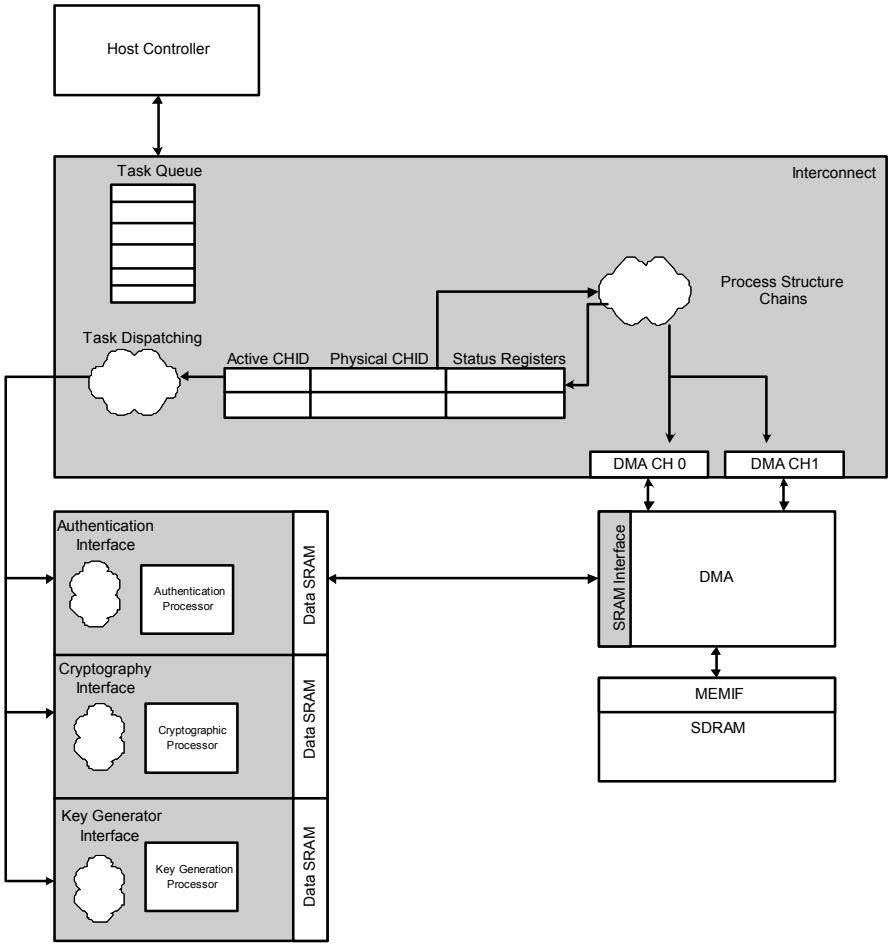
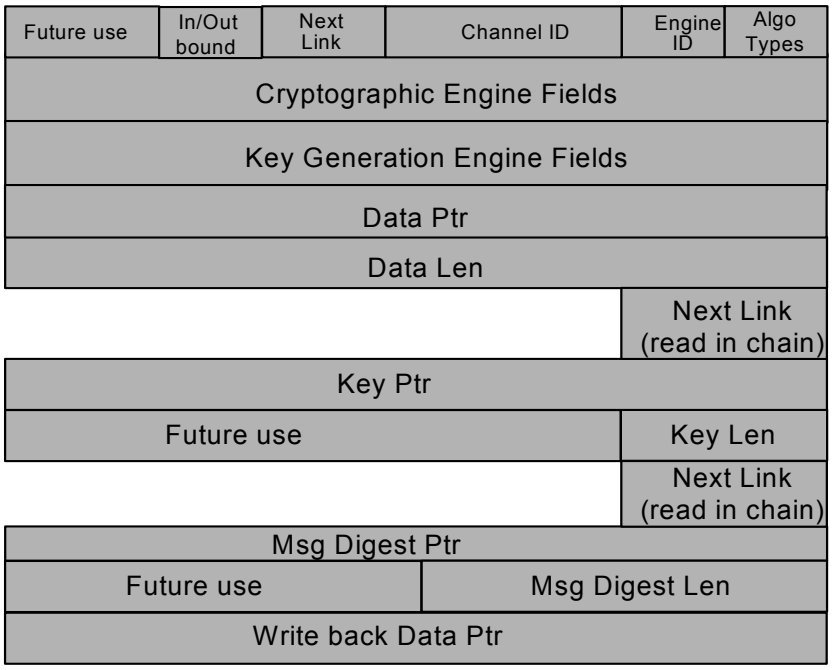


Figure 5: Interconnect TopLevel Architecture

The following InterConnect task structure indicates different fields for cipher, authentication and key engines.



5 INTERFACES

The Processor provides simplified interfaces with the host and external memory resource.

5.1 Host Interface

A 64-bit host interface is provided to allow the host to offload cryptographic functions to security processor task queue. Host can access program and data memories and all the registers of security processor through the host interface.

5.2 DMA Interface

DMA facilitates the functionality of the interconnect engine by bringing in and updating (as per requirement) the DMA structure resident in the external memory. The DMA has two channels and it carries out transfers between the external memory and local memories. DMA descriptors are specified in the structure read-in from the external memory, the specialized structure allows the programming of multiple DMA channels for processing different stages of a service.

6 RESULTS

Table 2 describes area and gate counts for different engines of the security processor. All numbers are with reference to 0.18µ CMOS Technology targeting 133MHz.

Component	Area	Gate Count
Cipher Engine	1087217.39	110K
Authentication Engine	1335479.62	134K
Key Generation Engine	1557018.12	158K
Interconnect & Interfaces	990718.95	100K
Total	4970434.08	502K

Table 2: Area & Gate Count

Table 3 lists the cycle required for different authentication, encryption, key generation, and signature generation and verification algorithms.

Algorithm	Block Size	Cycle Count
SHA-1	512-bit	160
SHA-256	512-bit	212
SHA-384	1024-bit	250
SHA-512	1024-bit	250
MD5	512-bit	205
DES	64-bit	6
3-DES	64-bit	18
AES-128	128-bit	20
AES-192	128-bit	24
AES-256	128-bit	28
DSA Signature Generation		295811
DSA Signature Verification		599097
Diffi-Hellmen Key		571244
RSA Private Key Operation		1211970
64-bit Random Number		120

Table 3: Cycle Counts

7 CONCLUSIONS

The layered re-configurable and programmable architecture provides scalable and modular hardware-software solution, in which components can be added/upgraded and removed as per requirements. A single layer of the processor provides a sustained throughput of up to 512Mbps.The processor can be upgraded to support Gigabit rates thus, making it an integral part of wire speed Network Processor. The programmable hardware can support future standards and other proprietary algorithms, which enhances the utilization of the ASIC. The security processor modules can be mapped on the same piece of silicon while porting on re-configurable architecture or FPGA. The software scheduler for optimally reconfiguring the FPGA logic with different security engines is currently being developed for the security processor and further research is being carried out for enhancing the performance in the re-configurable domain.

8 REFERENCES

[1] Maged Attia and Ingrid Verbauwhede, " Programmable Gigabit Ethernet Packet Processor Design Methodology ", ECCTD'01 - European Conference on Circuit Theory and Design, August 28-31, 2001, Espoo, Finland

[2] The MD5 Message-Digest Algorithm RFC1321

[3] Secure Hash Standard FIPS PUB 180-1

[4] Data Encryption Standard FIPS PUB 46-3

[5] Advanced Encryption Standard FIPS 197

[6] E. Sava¸s, C¸ K. Koc "The Montgomery Modular Inverse – Revisited" IEEE Transactions on Computers, Vol 49, No.7, July 2000.

[7] B.S.Kaliski Jr. "The Montgomery inverse and its applications". IEEE Transactions on Computers, 44(8):1064–1065, August 1995.