Cell Broadband Engine Support for Privacy, Security, and Digital Rights Management Applications

Kanna Shimizu
IBM Corporation
11501 Burnet Road
Austin, TX, 78758
512-838-1906
kannas@us.ibm.com

Daniel Brokenshire
IBM Corporation
11501 Burnet Road
Austin, TX, 78758
512-838-3373
brokensh@us.ibm.com

Mohammad Peyravian
IBM Corporation
3039 Cornwallis Drive
RTP, NC, 27709
919-481-4261
peyravn@us.ibm.com

ABSTRACT

The multi-core design of the Cell Broadband Engine Architecture (CBEA) presents an interesting opportunity for advancing secure computing. One class of cores on a CBEA chip, the Synergistic Processor Element (SPE), can be put into *isolation mode* whereby it is physically isolated from the rest of the system. Unlike many other proposed security architectures, this protection does not rely on any software mechanisms. Therefore, the scheme is robust against a compromised operating system or hypervisor, making CBEA uniquely attractive for security, privacy and digital content protection. Furthermore, the first implementation of this architecture, the Cell Broadband Engine (CBE), has produced compelling performance results for widely used cryptographic routines.

KEYWORDS

Cell Broadband Engine (CBE), Synergistic Processor Element (SPE), Security, Privacy, Digital Rights Management (DRM), Content Protection, AACS, Cryptographic implementation

INTRODUCTION

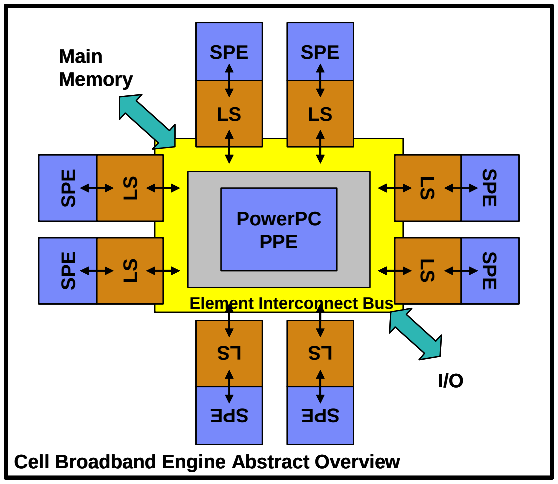
With the rapidly growing demand for stronger security, it is becoming increasingly clear that software alone cannot meet this need. Therefore, hardware, which is intrinsically less vulnerable to holes, manipulation, and attacks, must be re-thought and re-architected to support the security of the system. Only when hardware provides the right foundation can systems have a chance of providing effective security solutions. CBE is a general-purpose microprocessor designed with this goal in mind. It can, for example, support many of the features discussed in industry initiatives such as the Trusted Computing Group [TCG] and the Next Generation Secure Computing Base [MS]. However, what makes the CBE security architecture unique is that it does not solely rely on the integrity of the operating system or the hypervisor for security. It is designed such that *even if the operating system is compromised*, applications and data remain secure. This is in marked contrast with many other security architectures where once the operating system is compromised, all bets and guarantees are off.

In addition, CBE provides a hardware root of trust. It can be used at boot time to launch a trusted operating system and also at run time to launch a secure application. The hardware root of trust allows only trusted, un-tampered software to execute. Because this verification check is hardware-controlled, it cannot be manipulated by an adversary via ordinary means.

The organization of the paper is as follows: Section I presents an overview of CBE, Section II describes the SPE's isolation mode, Section III introduces the CBE's hardware root of trust, and Section IV presents performance results of CBE's cryptographic library routine. Section V will conclude the paper with a discussion of the possible applications of this architecture.

I. CELL BROADBAND ENGINE

The CBE has 9 processor cores which are connected via a high-bandwidth data ring (called the Element Interface Bus, EIB) [PAB05]. One of the cores, the 64-bit Power Processor Element (PPE), is the principal processor assuming a supervisory role. The 8 SPEs are the computational workhorses: RISC-style SIMD instruction set, wide and large (128 128 bit) register file, and 256Kbytes of physically dedicated private memory (for each SPE) [FAD05]. The SPE fetches instructions and load/stores data to and from this private memory, called the Local Store (LS). However, LS is not a hardware-managed cache; software is expected to explicitly transfer code and data into the LS. The transfers can happen to and from any resource on the EIB such as system memory, LS of other SPEs, and I/O devices. The main advantages of the LS are performance (coherency is not maintained for example) and area (circuitry for cache coherency can be quite large).

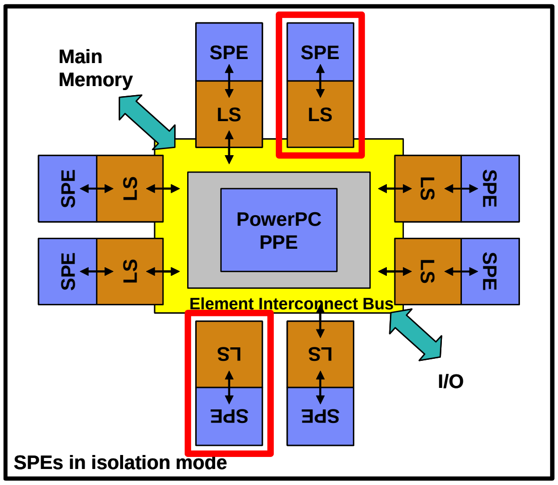


Cell Broadband Engine Abstract Overview

II. SPE ISOLATION MODE

At the heart of CBE's security architecture is the ability to isolate an SPE from the rest of the system. This is accomplished by one, locking up the isolated SPE's LS for its own use only, and two, disabling all external execution path control of the SPE core. Specifically, all LS read and write requests originating from units on the EIB such as the PPE, other SPEs, and the I/O do not have any effect on the locked-up region of the LS. There is a small area of the LS left open to both the external agents and the SPE for communication purposes. And, the isolation mode disables the ability for a supervisory process to set or read the program counter of the SPE. Once the SPE is isolated, the only external action possible is to cancel its task, whereby all data in the LS and SPE are erased before external access is re-enabled.

All of this is accomplished *exclusively* by hardware means; there is no software (in the form of setting protection bits in a table for example) involved in the process. Because of this absolute hardware isolation, even the operating system and the hypervisor cannot access the locked up LS or take control of the SPE core. Therefore, a hacker who has gained root or hypervisor privileges is not a threat to an application executing on an isolated SPE. The supervisory privileges will not enable him to control the application, nor will it allow him to read or write the memory used by it. The execution flow and the data of the isolated application are safe from manipulation, snooping, and modifications.



SPEs in isolation mode

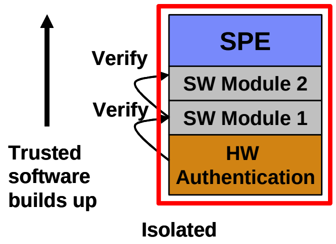
Using this capability, a CBE system can achieve the seemingly contradictory goal of protecting a user while simultaneously limiting that user's ability to infringe upon the security of others. Sensitive private information such as the users' passwords or credit card numbers can be designated to only be in the clear within an isolated SPE. At the same time, digital entertainment content can be protected from malicious users by always decrypting within an isolated SPE.

III. HARDWARE ROOT OF TRUST

Due to the malleability of software, it is generally believed that the root of an authentication scheme must be implemented in hardware. If the root can be trusted, then the entity authenticated by the root can be trusted, and so on as the chain of trust expands.

The CBE has a cryptographic-based hardware authentication mechanism that can be used as a foundation of a trusted software stack. The hardware verifies the integrity of the first software module, and in turn, this software module verifies the integrity of a secondary software module.

Because the hardware cannot be modified, its operation can be trusted and because the integrity of the first software module was verified by this trusted party, its operation can be trusted, and so on.



This hardware authentication mechanism is activated every time an SPE enters isolation mode. Specifically, the first code module to run during an isolated SPE session has its integrity checked by the hardware root of trust before it is executed. Thus, much like the two parts of an inductive proof, if the SPE isolation mode protects the software's execution, then, the hardware root of trust ensures that the software's initial state is correct.

IV. CRYPTOGRAPHIC LIBRARY

In addition to these architectural features, the CBE offers excellent performance for cryptographic functions. The large and wide SPE register file and the instruction set are highly amenable to the algorithms. The software team in the STI (Sony Toshiba IBM) Design Center has implemented several cryptographic standards for the SPE including symmetric key algorithms such as AES (Advanced Encryption Standard) and DES (Digital Encryption Standard), hash functions such as SHA-1 (Secure Hash Algorithm), and asymmetric key services such as RSA (Rivest-Shamir-Adleman), and DSA (Digital Signature Algorithm).

These functions were executed on CBE hardware in lab. In the following table, the second column lists the performance results for a single SPE on a CBE running at 3.2 GHz. The third column is the expected performance, based on the measured single SPE performance, when all 8 SPEs are used for functions that can be partitioned and executed in parallel. For example, AES in ECB (Electronic Code Book) mode or CTR (Counter) mode, where each plaintext block is encrypted independently, is a task that can be partitioned over multiple SPEs. The fourth column lists results for a leading brand processor using its commercially available cryptographic library implementation. The library is designed, implemented, and released by the manufacturer of the processor.

Symmetric Key Algorithms, Hash Algorithms

Function	1 SPE@ 3.2Ghz (Gbit/s)	8 SPE@ 3.2Ghz (Gbit/s)[1]	Leading Brand@ 3.2Ghz (Gbit/s)[2]
AES Encrypt (ECB)			
128 bit key	2.059	16.474	1.029
192 bit key	1.710	13.677	0.877
256 bit key	1.462	11.699	0.762
AES Decrypt (ECB)			
128 bit key	1.499	11.994	1.035
192 bit key	1.252	10.016	0.870
256 bit key	1.068	8.544	0.758
AES (CTR)			
128 bit key	1.966	15.725	Not Avail.
192 bit key	1.646	13.165	
256 bit key	1.415	11.322	
DES (ECB)	0.492	3.936	0.427
TDES (ECB)	0.174	1.395	0.133
SHA-1	2.116	N/A	0.902

AES in CTR mode has provably equal security as CBC (Cipher Block Chaining) mode, and protects against attacks which ECB mode is vulnerable to [Stal03]. In addition, CTR mode has superior performance compared to CBC mode because of its inherent parallelism [Stal03] and is very comparable to ECB mode performance (for 128-bit key encryption: 2.059 Gbits/s vs. 1.966 Gbits/s). Thus, the relative performance difference between the two processors for CTR mode is expected to be similar to the performance difference seen in ECB mode.

Asymmetric Key Algorithms

Function (For 1024 bit)	1 SPE (Mcycles)	Leading Brand (Mcycles)
Primality Test	2.665	15.09
Random Prime Gen.	827.0	1759
RSA Key Gen.	2217	3115
RSA Private Key En/Decryption	4.074	4.076
DSA Key Gen.	1331	7995
DSA Signing	2.250	2.887
DSA Verification	4.375	5.126

[1] Expected performance based on single SPE results
[2] Best-effort execution of vendor's code on hardware Results are linearly scaled to 3.2 GHz for comparison.

For the asymmetric key algorithms, the results for RSA and DSA are listed for 1024 bit keys. Note that the units used (Mega-cycles) are different from the AES performance measurements since in this case, the data size is fixed.

V. CONCLUSIONS & FUTURE WORK

By exploiting its unique architectural attributes of multiple independent cores and private Local Store, the Cell BE offers a security foundation that goes well beyond what can be delivered by other security architectures discussed today. For example, virtualization support which is receiving renewed attention as a hardware-based security solution [AMD05][Int03][MS] has been part of the IBM PowerPC® Architecture [IBM03] for quite some time and is fully incorporated in the Cell BE architecture although it is not thought of as a primary security offering. Also, although it proposes many useful approaches to trust establishment, a TCG-based strategy is vulnerable to basic and common security compromises such as buffer-overflow attacks [SVW04] which the Cell approach is resistant to. Therefore, Cell provides an essential building block for current and future secure systems.

The next step is to further develop software that would allow system designers and application programmers to fully exploit this architecture. The usage scenarios developed by them, such as the various content protection schemes, drive requirements on the lower-level software layers. Thus, there will be more focus on the development of standardized programming models and interfaces for the CBE security features.

REFERENCES

[TCG] Trusted Computing Group, https://www.trustedcomputinggroup.org/home

[MS] Microsoft NGSCB http://www.microsoft.com/resources/ngscb/

[PAB05] Pham et al. The Design and Implementation of a First-Generation CELL Processor. In *IEEE International Solid-State Circuits Symposium*, Feb. 2005.

[FAD05] B. Flachs et al. The Microarchitecture of the Streaming Processor for a CELL Processor. In *IEEE International Solid-State Circuits Symposium*, Feb. 2005.

[Stal03] William Stallings. Cryptography and Network Security, 2003.

[AMD05] AMD. AMD64 Virtualization Codenamed “Pacifica” Technology - Secure Virtual Machine Architecture Reference Manual, May 2005.

[Int03] Intel. LaGrande Technology Architectural Overview, September 2003.

[IBM03] IBM. PowerPC Operating Environment Architecture – Book III, Version 2.01, December 2003.

[SVW04] Reiner Sailer, Leendert Van Doorn, James P. Ward. The Role of TPM in Enterprise Security. In *IBM Research Report*, October, 2004.

IBM Corporation
Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
May 2005

This document was developed for products and/or services offered in the United States. IBM may not offer the products, features, or services discussed in this document in other countries.

The information may be subject to change without notice. Consult your local IBM business contact for information on the products, features and services available in your area.

All statements regarding IBM future directions and intent are subject to change or withdrawal without notice and represent goals and objectives only.

Photographs show engineering and design models. Changes may be incorporated in production models. Copying or downloading the images contained in this document is expressly prohibited without the written consent of IBM

All performance information was determined in a controlled environment. Actual results may vary. Performance information is provided “AS IS” and no warranties or guarantees are expressed or implied by IBM

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.