Mapping AES Cryptography and Whirlpool Hashing onto the Cell BE architecture

Roman Wyrzykowski, Lukasz Kuczynski, and Krzysztof Rojek

Institute of Computational and Information Sciences,
Czestochowa University of Technology
[roman,lkucz]@icis.pcz.pl

Abstract. The impressive computational power of the Cell Broadband Engine, coupled with its advanced security architecture, make it a perspective platform to implement cryptograpphic algorithms. This paper deals with mapping the AES symmetric-key cryptography and Whirlpool hash function onto the Cell Broadband Engine architecture.

Based on the analysis of possible approaches to mapping AES onto Cell BE architecture, we finally focus on two schemes. The first of them is based on using the efficient implementation of simultaneous table lookups in a 256-entry byte table stored in 16 vector registers, to perform the SubBytes transformations. In the second scheme, we merge SubBytes, ShiftRows, and MixColumns into a single shorter sequence of operations, which is implemented using table lookups in four 1KB tables stored in memory.

During the Conference, the performance results of the AES implementation on the Cell BE processor will be presented. In particular, we will report result of performance comparison of the first scheme, which is characterized by the extensive vectorization of operations on data stored entirely in registers, with the second scheme, which allows to decrease the number of operations at the cost of a lower degree of parallelism and longer time of access to data stored in the local memory of SPE.

Another topic of research carried out in this work is related to the efficient implementation of the cryptographic hash algorithms on the Cell BE. One of potential alternatives to the traditional solutions is a hash function called Whirlpool. The block cipher used by Whirlpool is very similar to the AES algorithm. This allows us to adopt for the Whirlpool algorithm the both of schemes employed for the AES algorithm. At the same time, the important difference between these algorithms is that AES operates of data viewed as 4-by-4 matrices of bytes, while in Whirpool two parallel datapaths operate on input data and key material, each viewed as 8-by-8 matrices of bytes. It gives us much wider possibilities for parallelizing the Whrilpool algorithm on the Cell BE processor, even for compressing a single block of data.

In the paper, for the first time, we propose how to decompose each datapath to provide its parallel execution on up to 4 SPEs. Such an approach would allow us to exploit efficiently the computing resources of all 8 SPEs in one Cell BE processor, even when processing a single stream of data.

1 Introduction

There is a rapid increase in sensitive data, such as biomedical records or financial data. Protecting such data while in transit as well as while at rest is crucial [8], [13]. Consequently, suitable techniques and tools, including cryptographic ones, should be applied to provide such key security properties as confidentiality, integrity, and authentication [4]. The impressive computational power of the Cell Broadband Engine [6], coupled with its advanced security architecture [11], make it a perspective platform to implement cryptographic algorithms [3], [12].

This paper deals with mapping two basic cryptographic algorithms [4], namely the symmetric-key cryptography and hash function, onto the Cell Broadband Engine architecture.

2 AES Symmetric Key Cryptography on Cell BE

The Advanced Encryption Standard (AES)[4], also known as Rijndael, is a block cipher adopted as an encryption standard by the U.S. government. It has been analyzed extensively and is now used worldwide.

A single round of AES consists of four steps:

1. SubBytes,
2. ShiftRows,
3. MixColums,
4. AddRoundKey.

A wide variety of approaches to implementing AES have been proposed, to satisfy the varying criteria of different applications and target architectures (see e.g. [5], [7]). Some performance results of implementing AES on the Cell BE are reported by K. Shimizu et al. [12], but practically without any details of their implementation.

In this paper, based on a systematic analysis of possible approaches to mapping AES onto Cell BE architecture, we finally focus on two schemes. The first of them is based on using the efficient implementation of simultaneous table lookups in a 256-entry byte table stored in 16 vector registers, to perform the SubBytes transformations. In the second scheme, we merge SubBytes, ShiftRows, and MixColumns into a single shorter sequence of operations [4], which is implemented using table lookups in four 1KB tables stored in memory. When implementing the second scheme, the level of parallelism is lower than in the first case.

For a single plaintext block, the AES computations have to be limited to a single SPE. By applying the CTR chaining mode for processing multiple blocks, it becomes possible to exploit computing resources of all 8 SPEs, when each plaintext block is encrypted or decrypted independently.

During the Conference, the performance results of the AES implementation on the Cell BE processor will be presented. In particular, we will report result of performance comparison of the first scheme, which is characterized by the extensive vectorization of operations on data stored entirely in registers, with

the second scheme, which allows to decrease the number of operations at the cost of a lower level of parallelism and longer time of access to data stored in the local memory of SPE.

3 Whirlpool Hashing Function on Cell BE

Another topic of research carried out in this work is related to the efficient implementation of the cryptographic hash algorithms on the Cell BE. These algorithms are employed [4] in numerous cryptographic applications. For example, they are widely used for message authentication, and digital signature.

A hash function takes in data with an arbitrary length and compresses them into a fixed-size output value called a hash or message digest. The required properties of cryptographic hash functions are that they are one-way functions and collision resistant [1], citebook.

Traditionally, SHA-1 and MD5 have been the most utilized functions. However, researchers have recently found weaknesses in both of them. This has raised a need for new designs. One of potential alternatives to the traditional solutions is a hash function called Whirlpool [2]. As originally submitted for the NESSIE European project, it is now adopted by the International Organization for Standardization (ISO/IEC) 101-18-3 standard. According to the authors' knowledge, only several publications concerning the implementation of Whirlpool on a concrete architecture have previously been published, and none of them consider the Cell BE architecture. For example, the hardware VLSI implementations using FPGAs have been investigated in [1], [9], and [10].

Whirlpool consists of the iterated application of a compression function, based on an underlying dedicated 512-bit block cipher that uses a 512-bit key. The block cipher used by Whirlpool is very similar to the AES algorithm. This allows us to adopt for the Whirlpool algorithm the both schemes employed for the AES algorithm.

At the same time, the important difference between the AES and Whirlpool algorithms is that AES operates of data viewed as 4-by-4 matrices of bytes, while in Whirpool two parallel datapaths operate on input data and key material, each viewed as 8-by-8 matrices of bytes. It gives us much wider possibilities for parallelizing the Whrilpool algorithm on the Cell BE architecture, even for compressing a single block of data.

First of all, the datapaths for input data and key material can be assigned to two different SPEs. Furthermore, for the first time, we propose how to decompose each datapath to provide its parallel execution on up to 4 SPEs. The possible approaches to this problem are based either on the geometrical decomposition of 8-by-8 input data (key material) matrices, or the following decomposition [2] of the circulant matrix $\mathbf{C}$ of order $2m = 8$, which describe the linear diffusion step of Whirlpool:

$$C = \begin{pmatrix} \mathbf{U} & \mathbf{V} \\ \mathbf{V} & \mathbf{U} \end{pmatrix},$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices of order $m = 4$. Such an innovative approach would allow us to exploit efficiently the computing resources of all 8 SPEs available in one Cell BE processor, even when processing a single stream of input data.

References

1. Alho, T., et al.: Compact Hardware Design of Whirlpool Hashing Core. Design, Automation & Test in Europe Conference & Exhibition, 2007
2. Barreto, P., Rijmen, V.: The Whirlpool Hashing fubction, http://paginas.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html
3. Costigan, S.: Accelerating SSL using the vector processors in IBMs Cell Broadband Engine for Sonys Playstation 3. In: SPEED:Software Performance Enhancement for Encryption and Decryption, Amsterdam, 2007, 65–76
4. Denis, T.S., Johnson, S.: Cryptography for Developers. Syngress Publishing, Rockland, MA (2007)
5. Fiskiran, A.M., Lee, R.B.: On-Chip Lookup Tables for Fast Symmetric-Key Encryption. Proc. IEEE 16th Int. Conf. Application-Specific Systems, Architectures, and Processors (ASAP'06), IEEE Computer Society, 2006
6. Gschwind, M., et al.: Synergistic Processing in Cell's Multicore Architecture. IEEE Micro, No.2 (2006) 10–24
7. Irwin, J., Page D.: Using Media Processors for Low-Memory AES Implementation. Proc. IEEE 13th Int. Conf. Application-Specific Systems, Architectures, and Processors (ASAP'03), IEEE Computer Society, 2003
8. Kher, V., Kim, Y.: Securing Distributed Storage: Challenges, Techniques, and Systems. Proc. 2005 ACM workshop on Storage Security and Survivability, Fairfax, VA, USA, 2005, 9–25
9. Kitsos, P., Koufopavlou, O.: Efficient Architecture and Hardware Implementation of the Whirlpool Hash Function. IEEE Trans. on Consumer Electronics **50**, **1** (Feb. 2004) 208–213
10. McLoone, M., McIvor C., Savage, A.: High-Speed Architectures of the Whirlpool Hash Function. Proc. 2005 IEEE Int. Conf. on Field-Programmable Technology, 2005
11. Shimizu, K.: The Cell Broadband Engine processor security architecture, http://www-128.ibm.com/developerworks/power/library/pa-cellsecurity/
12. Shimizu, K., Brokenshire, D., Peyravian, M.: Cell Broadband Engine Support for Privacy, Security, and Digital Rights Management Applications, http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/
13. Wyrzykowski, R., Kuczynski, L.: Towards Secure Data Management System for Grid Environment Based on the Cell Broadband Engine. Lect. Notes in Comp. Sci. **4967** (2008) (to appear)