Hardware Acceleration for Cryptographic Functions

(AES Algorithm)

By: Ahmed Moussa
Outline

- Introduction
  - Why Accelerate Cryptographic Functions?
  - Why Hardware Acceleration?

- Approaches

- Literature Review

- Problem Statement
Why is acceleration is needed for cryptography?

• **Better performing web servers**

**SSL (Secure Socket Layer)** – cryptographic protocol used by majority of today’s web servers to protect and secure communications from and to the websites

- Estimated number of Web servers currently exceeds 285 million (*Netcraft Web Survey, February 2011*)
- Many of them face availability issues at high traffic times
Why is acceleration is needed for cryptography?

- Virtual Private Networks (VPNs)

**IPSec (Secure Internet Protocol)** – cryptographic protocol used to support VPNs (Virtual Private Networks), i.e., secure communication between remote Local Area Networks (LANs) using Internet.

IPSec optional in IP ver. 4, required in emerging IP ver. 6.
Why is acceleration is needed for cryptography?

- Virtual Private Networks (VPNs)
Why is acceleration is needed for cryptography?

• Need for high-end VPN devices
e.g. corporate security gateways and routers
  - speeds reaching 10 Gbit/s and beyond
  - delay & bandwidth sensitive applications:
    - Banking Transactions
    - VoIP (Voice over IP)
    - Video conferencing
Why is acceleration is needed for cryptography?

- **Storage Area Networks**
  Encryption of data during transmission and at rest.

- **Pay TV**
  - High volume
  - Pay TV decoders must be tamper-resistant
  - Capability of a remote upgrade can substantially reduce the cost of recovering from an attack
Why Hardware Acceleration?

Offload some heavy lifting to a different unit!
Why Hardware Acceleration?

- **Co. Processors VS Accelerator**
  - A *co-processor* connects to the internals of the CPU and executes instructions.
    - Instructions are dispatched by the CPU.
  - An *accelerator* appears as a device on the bus.
    - Accelerator is controlled by registers, just like I/O devices
    - CPU and accelerator may also communicate via shared memory, using synchronization mechanisms
    - Designed to perform a specific function
Why Hardware Acceleration?

- **Accelerated system architecture**

![Diagram]

- CPU
- Accelerator
- Memory
- I/O

- Request data
- Result data
Why Hardware Acceleration?

• Better cost/performance.
  – Custom logic may be able to perform operation faster than a CPU of equivalent cost.
  – CPU cost is a non-linear function of performance.
    - Hence, better split application on multiple cheaper execution units
Why Hardware Acceleration?

- Good for processing I/O in real-time.
- May consume less energy.
- May be better at streaming data.
- May not be able to do all the work on even the largest single CPU.
Approaches for H.W. Acceleration

• Application-specific integrated circuit (ASIC).

• Field-programmable gate array (FPGA).

• Standard component.
  – Example: graphics processor unit (GPU).
Why GPUs?

- Power Consumption
- Performance per $
- Multi-usage

- The authors here investigate GPU acceleration of symmetric-key and asymmetric-key functions, fundamental components of modern cryptographic systems.

- They show that AES, a popular example of a symmetric-key function, can be competitive with the CPU on recent GPUs and outperform on contemporary GPUs.

- They also illustrate the issues related to GPU support of symmetric-key modes of operations in various scenarios and present strategies for maintaining performance.
Literature Review

2) **GPU accelerated cryptography as an OS service**, Harrison, O., & Waldron, J., Springer Transactions on Computational Science XI (2010).

The authors here acknowledges the research already done before in usage of GPUs as a crypto accelerator and achieving better results than CPUs

- Despite the existence of these new approaches, there remains no way for OS kernel services or user’s pace applications to make use of these implementations in a practical manner.

- To overcome this shortcoming, this paper investigates the integration of GPU accelerated cryptographic algorithms with an established service virtualization layer within the Linux kernel, the OCF-Linux framework.
Literature Review

2) **GPU accelerated cryptography as an OS service,** Harrison, O., & Waldron, J., Springer Transactions on Computational Science XI (2010).

- The authors here acknowledges that a traditional AES CUDA implementation does not specify efficient techniques to utilize the GPU parallelism. As a result, in this research, we evaluate the recent proposed parallel AES implementations over GPU.

- In addition, to avoid the frequent shared memory access, we offer the possibility to rearrange AES stages to utilize the AES parallelism resulting in a momentous improvement of encryption speed-up over traditional GPU implementation.
Future work

• The authors in [2] provided an architecture for GPU accelerated cryptography as an OS service to be added to the Open Cryptographic Framework however no implementations were provided although the idea is very promising and quiet an addition to OCF.

• The work done in [1] & [3] and most of the recent work in terms of acceleration of AES on GPUs doesn’t make use of the modern functionalities and CUDA framework features due to the usage of outdated hardware in their experiments.

• More investigations should be done towards other cryptographic algorithms other than AES.

• Comparative studies should be done over applications performance with and without the use of acceleration as not all cryptographic function may suit the parallel architecture of GPUs (for ex. Chaining algorithms)
Reference


2- GPU accelerated cryptography as an OS service, Harrison, O., & Waldron, J., Springer Transactions on Computational Science XI (2010).


5- Michael Kipper, Joshua Slavkin, Dmitry Denisenko, “AES On GPU”, University of Toronto,2009

6- AES on GPU, http://www.eecg.toronto.edu/~moshovos/CUDA08/doku.php?id=project_presentations_reports_source_code#aes_on_gpu