Abstract

Public key algorithms are extensively known to be slower than symmetric key alternatives in the area of cryptographic algorithms for the reason of their basis in modular arithmetic. The most public key algorithm widely used is the RSA. Therefore, how to enhance the speed of RSA algorithm has been the research significant topic in the computer security as well as in computing fields. With remarkable increase in the computing capability of the modern Graphics Processing Unit's (GPUs) as a co-processor of the CPU, one can significantly benefit from the Single Instruction Multiple Thread (SIMT) style of computing. This paper proposes a hybrid system to parallelize the RSA for multicore CPU and many cores GPUs with variable key size. In doing so, three variants implementation for the RSA algorithm are done to facilitate the performance comparison against Crypto++ library and sequential counterpart. The GPU implementation gained approximately 23 speed up factor over the sequential CPU implementation; while the multithread CPU implementation gained only 6 speed up factor over the sequential CPU implementation as far as the latency is concerned. Furthermore, additional speedup could be gained as far as the throughput is concerned; the throughput gained for 1024 bits is ~1800 msg/sec; as for 2048 bits is ~250 msg/sec. Due to overlapping of multithread operation whenever free resources are available. The experiments are conducted on a laptop with Intel Core i7-2670QM, 2. 20 GHz CPU and Nvidia GeForce GT630M GPU. Results reveal
that the GPU is appropriate to speed up the RSA algorithm.

References

- Harrison, O. and Waldron, J. "Efficient Acceleration of Asymmetric Cryptography on Graphics Hardware"; The 2nd International Conference on Cryptology in Africa,
Parallelizing RSA Algorithm on Multicore CPU and GPU


- CUDA C PROGRAMMING GUIDE, 2013, NVIDIA.

**Index Terms**

Computer Science

Security
Keywords
RSA; SIMT; GPU; Parallel algorithms; Heterogeneous computing.