Hardware-Assisted Security Mechanism: the Acceleration of Cryptographic Operations with Low Hardware Cost

Jed Kao-Tung Chang
Shaoshan Liu
Jean-Luc Gaudiot
Dept of Electrical Engineering and Computer Science
University of California
Irvine, CA 92697, USA
{jedc, shaoshal, gaudiot}@uci.edu

Chen Liu
Department of Electrical and Computer Engineering
Florida International University
Miami, FL 33174, USA
cliu@fiu.edu

I. INTRODUCTION

In the past 30 years, more and more information is stored in digitized format due to the rapid development of computer and communication technology. However, without good protection scheme, data will be hacked and cracked by the malicious intruders. A good security mechanism will keep information secrecy and integrity. However, cryptography algorithms are extremely expensive in terms of execution time because many arithmetic and logic operations are executed in the encryption/decryption process to make data not easily cracked. Huge amount of data also are transmitted between the CPU and memory. Using a traditional general-purpose processor would not be efficient for this scenario. Thus, the performance enhancement for the security system is crucial in modern world.

To address this issue, we expedite the cryptographic operations through hardware acceleration. Compared with many previous works which focus on the algorithm design or the parallelization technique, we would like to provide a generic cryptographic accelerator. We found there are certain "hotspot functions" in the cryptographic algorithm which consume a substantial amount of the execution time of the specific algorithm. By moving the operations to hardware, we can reduce the overheads introduced by the crypto-computation so that the computing resource can focus on the useful work. To this end, we hope this will provide hardware-assisted solutions to improve the performance of the security mechanism of modern enterprise and IT systems.

II. CANDIDATE ALGORITHMS

We have collected nine benchmarks used widely in modern security systems: AES, 3DES, RC5, MD5, IDEA, SHA1, Blowfish, ECC and RSA. The reason we choose these nine benchmarks is due to their popularity and their program structures (Feistel Cipher and Iterated Block Cipher), which represent the ability of contemporary cryptography works. Rijndael's algorithm has been selected as AES due to its good balance in terms of speed, security, and flexibility. 3DES applied the DES three times to each data block and is widely adopted in banking information system and electronic payment industry. IDEA is used by PGP (Pretty Good Privacy) v2.0 to transmit message bodies. Blowfish provides a good encryption rate which takes 18 cycles to encrypt one byte on a 32-bit processor. Its memory requirement is only 5KB. ECC and RSA are public key algorithms. ECC is based on the algebraic structure of elliptic curves over finite fields. It is now popular due to the fact that it offers the same security level as offered by other contemporary algorithms at a shorter key length. RSA is suitable for encryption and digital signature and used in E-Commerce protocols. Although the execution of ECC and RSA are time-consuming, in these asymmetric algorithms each data can be encrypted or decrypted independently and the operations on these data can be performed in parallel. RC5, MD5, and SHA1 are all hash algorithms, which are used to verify the integrity of data blocks. RC5 is notable for its simplicity. What's more, the length of its secret key, word size, and number of rounds of computation can be varied. It is used in devices with restricted memory size such as smart cards. MD5 is widely used to assure if the transmitted file has arrived intact and to store passwords. SHA1 is often used in firewall, VPN, and IP-security.

III. HARDWARE ACCELERATION

Our approach consists of two steps. The first step is to identify the hotspot functions. Then we convert these functions in hardware accelerators in the second step.

In the first step, two aspects of the hotspot function are worth considering. We need to choose a hotspot function with high execution rate. If the rate is too low, implementing this function would make little sense. The hardware cost to implement hotspot function is also important. In some cryptographic algorithms the hotspot function is exactly the main process of the total algorithm, such as the encryption/decryption part. The hardware cost will be too high and much die area will be consumed for such cases. Thus a hotspot function with high execution rate and low hardware cost will be suitable for hardware acceleration.

We used the INTEL performance analyzer VTune, which analyzes the software performance on IA-32 and Intel64-based machines, to examine the hotspot function. From the call graph (showing the call relationship among all functions) and the execution time of each function, we identify the hotspot function(s) of each benchmark. SHA1's hotspot function has low execution rate. Other algorithms' hotspot functions are

engaged in crypto-computations and occupy most of the work. They are not good object for hardware acceleration due to the high hardware cost. Based on our observation, RSA and AES are suitable for hardware acceleration since their hotspot functions have high execution rate and hardware costs are low. Next we implemented the hotspot functions of the selected benchmarks on Xilinx Virtex-5 FPGA board at 100MHz. We achieve 30-100 folds performance improvement compared to the pure software implementation of the hotspot functions. In the meantime, we achieve the speedups of 2.9 for RSA, 4.5 for AES Encryption, and 5.9 for AES Decryption in terms of overall execution time of the cryptography algorithms.

IV. CONCLUSION

Hardware acceleration for cryptographic algorithms not only enhances the performance of the security systems but also leaves the computing resource dedicating on more useful work. In the future we will look further into the hotspot function to see if there is certain “hot line” or “hot block” which takes a significant amount of execution time and perform the hardware acceleration at a even finer granularity.