Chapter 12
Architectural Support for Enhancing Critical Secrets Protection in Chip-Multiprocessors

Lu Peng
Louisiana State University, USA

Li Yang
University of Tennessee at Chattanooga, USA

Balachandran Ramadass
Louisiana State University, USA

ABSTRACT

Security has been considered as an important issue in processor design. Most of the existing designs of security handling assume the chip as a single secure unit. However, such assumption is vulnerable to exposure resulted from a central failure point. This chapter proposes a secure Chip-Multiprocessor architecture (SecCMP) to handle security related problems such as key protection and core authentication in multi-core systems. Matching the nature of multi-core systems, a distributed threshold secret sharing scheme is employed to protect critical secrets. A critical secret (e.g., encryption key) is divided into multiple shares and distributed among multiple cores instead of being kept a single copy in one core that is sensitive to exposure. The proposed SecCMP can not only enhance the security and fault-tolerance in secret protection but also support core authentication. SecCMP is designed to be an efficient and secure architecture for CMPs.

INTRODUCTION

Computer networking makes every computer component vulnerable to security attacks. Examples of such attacks include injection of malicious codes (e.g., buffer overflow), denial of service (DoS) attacks, and passive eavesdropping between CPU cores and off-chip devices. Also off-chip or on-chip devices taken over by an adversary can launch attacks to other components of a computer. Pure software solutions itself can not counter all attacks,

DOI: 10.4018/978-1-61692-000-5.ch012

therefore, enforcing security in processor design has drawn more and more attention. Currently many proposed works focus on encryption and authentication of hardware memory in single-core systems (Gassend, Suh, Clarke, Dijk, & Devadas, 2003; Lee, Kwan, McGregor, Dwoskin, & Wang, 2005; Shi, Lee, Ghosh, Lu, & Boldyreva, 2005; Yan, Rogers, Englender, Solihin, & Prvulovic, 2006; Yang, Zhang, & Gao, 2003). They usually assume the processor core as a safe and secure unit. When Chip-Multiprocessors (CMPs) have become mainstream products, applying encryption scheme of existing works to each core independently is one possible solution to enforce security in CMPs. The weakness of this solution is that the critical secrets (e.g. encryption key) stored or processed by one processor core can be easily exposed to adversaries through remote exploit attacks such as buffer overflow or Trojan horse, which leads to a central failure point. Once a core is compromised or taken over, the adversary could either access the critical secrets or wait until the compromised thread migrating onto another clean core then access unauthorized critical secrets. Therefore, this is not an effective approach to protecting shared critical secrets for CMPs.

Utilizing the distributed nature of CMPs is an alternative solution to reinforce the security of CMPs. Not only the computation load but also the security risks are distributed among multiple processor cores that are designed to collaboratively protect and access the critical secret. No individual core is possible to access the critical secret alone. We proposed a novel *Secure Chip-Multiprocessor (SecCMP)* architecture (Yang, & Peng 2006) to protect critical secrets based on a distributed *Secret Sharing* (Pedersen 1991). Instead of protecting a secret in one processor core, *Secret Sharing* is employed to distribute the secret among multiple cores that protect the secret collaboratively. The distributed security management matches the nature of multi-core architecture in CMPs. By employing a threshold *Secret Sharing* scheme, critical secrets are protected safely in a CMP processor even when one or more processor cores are compromised. In this chapter we integrate the SecCMP architecture with identity based cryptography to support remote information access and sharing. The performance degradation of our approach is studied through simulation. Low overheads and improved fault-tolerance are two major features of our approach. Low overhead is achieved via distributing the encryption and decryption load among multiple cores. Fault-tolerant is achieved via *(k, n)* secret sharing where at least k out of n cores are required to recover the secret. From a secret protection point of view, fewer than *k-1* cores are not able to recover the secret (i.e., the encryption key) such that our solution is resistant to the compromise of fewer than *k-1* cores. From a service protection point of view, k cores are able to provide the secret recovery service (i.e., retrieve the encryption key) such that our solution is tolerant to failure (i.e., hardware failure, DoS attacks) of up to *(n-k)* cores. Moreover, confidentiality and authentication among cores are supported through core authentication in *SecCMP*. Core authentication, which identifies whether a core is compromised, could be performed during critical information reconstruction or periodically. If not enough authenticated cores available, a system error will be called. The user may restart the system and reconstruct the critical secrets.

We use an application to demonstrate secure and remote critical information access and sharing supported by our *SecCMP*. Integrated with identity based cryptography (Bonh, & Franklin, 2003), the *SecCMP* provides a secure and reliable way to generate and distribute encryption keys between local host and remote site when prior distribution of keys is not available. Each local host has a pair of *master public key* (*MUK*) and *master private key* (*MRK*). In addition, each account has a pair of *account public key* (*AUK*) and *account private key* (*ARK*). In the local host which contains a multi-core processor, the MRK is divided and distributed among multiple cores and the ARK is generated from the MRK. On the