Hardware-Software Codesign of Pairing-Based Cryptosystems for Optimal Performance vs. Flexibility Trade-off

Malik Umar Sharif, Marcin Rogawski, and Kris Gaj
George Mason University

Abstract

One of the most promising directions in the theory and practice of computer and network security is the emergence of pairing-based cryptography (PBC). Developed by a group of researchers in early 2000s, this new approach to cryptography tackles problems beyond the reach of traditional cryptographic schemes, and makes many well-known cryptographic solutions less costly, less cumbersome, and easier to deploy. For a start, PBC enables a non-interactive key agreement, in which two parties can agree on a joint secret key without ever exchanging any information through either public or private channel. Even more importantly, identity-based encryption (IBE), enabled by pairing transformations, holds a promise of finally solving the centuries-old problem of key management. Other important applications include one-round tri-partite key agreement, attribute based encryption, secret handshake, as well as blind, proxy, and group signatures. Pairing-schemes over prime fields, such as Optimal Ate Pairing over Barreto-Naehrig (BN) curves, are considered particularly resistant to cryptanalysis, but at the same time, the most challenging to implement efficiently.

Traditionally, software implementations of pairing-based algorithms provided the highest flexibility but lacked performance. On the contrary, custom hardware accelerators provided the highest performance but lacked flexibility and adaptability to changing algorithms, parameters, and key sizes.

In this paper, we present a new design methodology, based on the hardware-software codesign, aimed at maximizing performance without sacrificing flexibility. Our platform is Xilinx Zynq-7000 SoC, which integrates a dual-core ARM Cortex-A9 processing system with a 28nm Xilinx programmable logic (equivalent to Artix-7 FPGA). The software part of our implementation is based on the RELIC library (Efficient Library for Cryptography), developed in Brazil as a part of the TinyPBC project, and optimized for embedded applications. The hardware part utilizes the Orup-Suzuki Montgomery multiplier based on DSP units of modern FPGAs. The data is exchanged between the ARM processing system and the hardware accelerator using a DMA engine, and the execution time is measured accurately using a hardware counter implemented in reconfigurable logic. The experimental testing is performed using Zynq Evaluation and Development Board – ZedBoard.

The performance vs. flexibility trade-off is investigated by changing the possible boundaries between the software and hardware part of our implementation, and moving an increasingly complex operation to hardware.