Abstract

There is a re-emerging demand for low-end devices such as 8-bit processors, driven by needs for pervasive applications like sensor networks and RF-ID tags. Security in pervasive applications, however, has been a major concern for their widespread acceptance. Public-key cryptosystems (PKC) like RSA and DSA generally involve computation-intensive arithmetic operations with operand sizes of $1024 - 2048$ bits, making them impractical on such constrained devices.

Elliptic Curve Cryptography (ECC) which has emerged as a viable alternative is a favored public-key cryptosystem for embedded systems due to its small key size, smaller operand length, and comparably low arithmetic requirements. However, implementing full-size, standardized ECC on 8-bit processors is still a major challenge and normally considered to be impracticable for small devices which are constrained in memory and computational power.

The thesis at hand is a step towards showing the practicability of PKC and in particular ECC on constrained devices. We leverage the flexibility that ECC provides with the different choices for parameters and algorithms at different hierarchies of the implementation. First a secure key exchange using PKC on a low-end wireless device with the computational power of a widely used 8-bit 8051 processor is presented. An *Elliptic Curve Diffie-Hellman* (ECDH) protocol is implemented over 131-bit Optimal Extension Field (OEF) purely in software. A secure end-to-end connection in an acceptable time of 3 seconds is shown to be possible on such constrained devices without requiring a cryptographic coprocessor.

We also investigate the potential of software/hardware co-design for architectural enhancements including instruction set extensions for low-level arithmetic used in ECC, most notably to speed-up multiplication in the finite fields. We show that a standard compliant 163-bit point multiplication can be computed in 0.113 sec on an 8-bit AVR micro-controller running at 4 Mhz (a typical representative for a low-cost pervasive processor) with minimal additional hardware extensions. Our design not only accelerates the computation by a factor of more than 30 compared to a software-only solution, it also reduces the code-size and data-RAM. Two new custom instructions for the MIPS 32-bit processor architecture are also proposed to accelerate the reduction modulo a pseudo Mersenne prime. We also show that the efficiency of multiplication in an OEF can be improved by a modified multiply and accumulate unit with a wider accumulator. The proposed architectural enhancements achieve a speed-up factor of 1.8 on the MIPS processor.

In addition, different architectural enhancements and optimal digit-size choices for the Least Significant Digit (LSD) multiplier for binary fields are presented. The two different architectures, the Double Accumulator Multiplier (DAM) and N-Accumulator Multiplier (NAM) are both faster compared to traditional LSD multipliers.

Later, an area/time efficient ECC processor architecture (for the OEFs of size 169, 289 and 361 bits) which performs all finite field arithmetic operations in the discrete Fourier domain is described. We show that a small optimized implementation of ECC processor with 24k equivalent gates on a 0.35um CMOS process can be realized for 169-bit curve using the novel multiplier design. Finally we also present a highly area optimized ASIC implementation of the ECC processor for various standard compliant binary curves ranging from $133 - 193$ bits. An area between 10k and 18k gates on a 0.35um CMOS process is possible for the different curves which makes the design very attractive for enabling ECC in constrained devices.