

Low-energy Implementation of ECDSA using Hardware Acceleration

F. Valencia, D. Besse, J.-L. Nagel

Through successful co-optimisation of hardware and software, CSEM designed and implemented high performance ECDSA security algorithms for digital signatures with low footprint and power consumption, suitable for embedded systems.

Security is a required property of all connected systems, although too often neglected on embedded systems due to the high incurred costs. Even if they go unnoticed, embedded systems manage more critical functions, use more sensible data, fulfill higher regulations and are more connected. CSEM implemented a high- performance, low footprint and low power acceleration of ECDSA, a digital signature algorithm based on elliptic curves, using a HW/SW codesign strategy. ECDSA C implementation was mapped to a RISC-V processor and a big-integer Montgomery arithmetic accelerator. The execution time was reduced by a factor of more than 12x while the active power only grows by 1.4x, yielding an energy reduction of 10x.

Elliptic Curve Cryptography (ECC) [1] is a family of Public Key Cryptography (PKE) based on elliptic curves over finite fields. ECC is very suitable for constrained devices because it has the smallest overhead in ciphertext/signature size and computational complexity of existing PKE algorithms. ECC can be used for key agreement, digital signatures, pseudo-random generators, etc. Elliptic curve operations heavily rely on big integer modulo arithmetic.

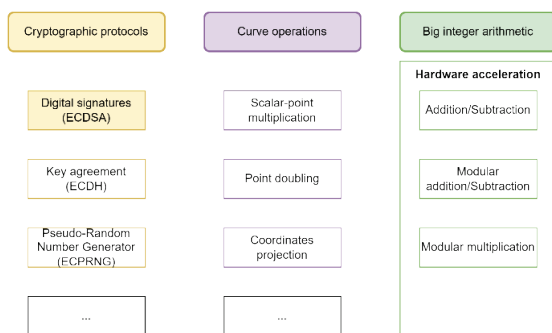


Figure 1: ECC layers.

Figure 1 shows three abstract layers in ECC protocols, where the lowest layer is the big integer arithmetic. This work presents the achieved improvements in ECDSA when big integer arithmetic is accelerated in hardware. The accelerator is integrated with a RISC-V processor. The uECC library [2] was used as reference.

Firstly, the uECC code was modified to perform the core operations in the Montgomery domain, which is known to be very efficient in modulo arithmetic. Adding conversions to/from Montgomery domain yields even more Montgomery multiplications as the transformation consists in multiplying by a precalculated constant.

In a second phase, all these multiplications were mapped to an accelerator IP which integrates to the SoC system [3] via the APB

bus, to send commands and configuration, and a small DMA to handle the data directly in the RAM. Not only does the accelerator perform the Montgomery computation faster than the processor, but also less bus transactions to transfer data back-and-forth are required.

For reusability, the accelerator is not specific to curve cryptography but implements generic Montgomery-domain big integer arithmetic. Primarily, it has been optimized for the Montgomery modular multiplication which accounts for the most operations. Nevertheless, the accelerator also supports addition and subtraction with and without modular reduction through reuse of the same hardware. For those operations, the gain in computation time is not as significant but supporting them allows to keep the operands inside the accelerator instead of transferring them to the processor.

A complete system combining CSEM's IcyFlex-V RISC-V processor with a 256-bit wide operands accelerator was simulated in GF22 library with a clock at 25MHz. The operands width is configurable though and clock frequencies up to a few hundreds of MHz were achieved in synthesis with an area of 17kGE for the accelerator.

Table 1 Comparison of ECDSA functions using hardware acceleration.

	SW	SW+acc	Improvement
Time key generation	570 ms	39 ms	14.3x
Time signing	612 ms	41 ms	14.3x
Time verification	696 ms	55 ms	12.5x
Power	273 μ W	385 μ W	0.7x
Energy	512 μ J	52 μ J	10x

The execution time of the three main function of ECDSA (key generation, signing and verification) is reduced by more than 12x. Even if the total power (processor and accelerator) increases by 1.4x, the energy is reduced by one order of magnitude thanks to the shorter execution time.

This work shows how important it is to co-optimize the software and hardware of an SoC to achieve efficient implementations of cryptographic protocols, as energy can be reduced by efficient hardware computation blocks and by minimizing the system load, both processor and bus. Generic accelerators will provide more agility and reuse possibilities when implementing similar approaches for newer cryptographic protocols such as the recently standardized post-quantum algorithms.

[1] Digital Signature Standard (DSS), National Institute of Standards and Technology (2023)

[2] Comparative Study of ECC Libraries for Embedded Devices, Silde, Tjerand (2019)

[3] J.-L. Nagel, Icyflex-V: a new ultra-low power processor based on RISC-V architecture, [CSEM Scientific and Technical Report \(2019\), 116](#)