Hardware Architectures Exploration for Hyper-Elliptic Curve Cryptography

Gabriel Gallin, Arnaud Tisserand

Hardware Architectures Exploration for Hyper-Elliptic Curve Cryptography *

Gabriel GALLIN[1,2] and Arnaud TISSERAND[1,3]
[1]CNRS – [2]IRISA UMR 6074, [3]Lab-STICC UMR 6285

Nowadays, there is an increasing number of applications and systems requiring strong security on small devices. Public-key cryptography (PKC) is mandatory for providing key exchange and digital signature. The first standard for public-key crypto-systems was RSA. However, to be compliant with the recommended theoretical security levels, RSA based crypto-systems must use large keys – at least two thousand bits – which makes it too costly for embedded applications.
Curves based cryptography such as *Elliptic Curve Cryptography* (ECC) or *Hyper-Elliptic Curve Cryptography* (HECC) is known to provide a given security level at a lower cost than RSA. For instance, 226-bit ECC keys offer the same security level as 2048-bit RSA. Due to its reduced cost and better performance, ECC is now recommended as the standard for public-key crypto-systems.

Recent research has pointed out HECC as an attractive alternative to ECC. HECC is based on a different kind of curves, which allows the size of the computed data to be halved, but at the expense of an increased number of finite field operations. In [1], Renes *et al.* present software implementations of key exchange and signature schemes based on HECC and Kummer surfaces, targeting embedded processors (*ARM Cortex M0* and *AVR ATmega*). The provided results show very interesting speedups compared to state-of-the-art ECC: 30% speedup for Diffie-Hellman key exchange and up to 70% for signature.
As pointed out above, operations on HECC involve more operations on the underlying finite field than ECC. However, one can observe that in ECC, most of the computations are dependent and must be mostly done in a sequential way. For this reason, the internal parallelism in ECC is quite limited compared to HECC. For instance, in the formulas presented in [1], one can find regular patterns of four to eight independent modular multiplications – the most costly and common finite field operation – feasible in parallel. HECC internal parallelism brings forward numerous questions for hardware implementations. Those questions can be summarized as follows: *how can one take advantage of the parallelism of HECC to design efficient hardware crypto-systems?*

Our research group has been studying arithmetic operators and implementations of hardware accelerators for ECC, with robustness against physical attacks such as *Side Channel Analysis* (SCA) or faults injections. We are now designing hardware accelerators for HECC scalar multiplication by exploring different types of architectures. We developed a specific CABA (Cycle Accurate, Bit Accurate) simulator for our architectures. With this simulator, we can study the impact of the type, number and size of the arithmetic units and of the choice between different types of parallel architecture on the performances, circuit area and resistance against physical attacks. We will also compare different ways to manage internal data transfers and different control flow implementations. The most interesting configurations will be implemented on FPGA and evaluated on our attack setup.

References

[1] J. Renes, P. Schwabe, B. Smith, and L. Batina, "μKummer: Efficient hyperelliptic signatures and key exchange on microcontrollers," in *Proc. Workshop on Cryptographic Hardware and Embedded Systems – CHES*, vol. 9813 of *LNCS*, pp. 301–320, Springer, Aug. 2016.

*This work is partly funded by the HAH project (Labex CominLab and Lebesgue, Brittany Region, `http://h-a-h.inria.fr/`).