High-Performance Elliptic Curve Cryptography: A SIMD Approach to Modern Curves
(Thesis Summary)[*]

Armando Faz-Hernandez, Julio López

Institute of Computing, University of Campinas.
1251 Albert Einstein, Cidade Universitária. Campinas, São Paulo. Brasil.
{armfazh,jlopez}@ic.unicamp.br

Abstract. *Cryptography based on elliptic curves is endowed with efficient methods for public-key cryptography. Recent research has shown the superiority of the Montgomery and Edwards curves over the Weierstrass curves as they require fewer arithmetic operations. Using these modern curves has, however, introduced several challenges to the cryptographic algorithm's design, opening up new opportunities for optimization.*

Our main objective is to propose algorithmic optimizations and implementation techniques for cryptographic algorithms based on elliptic curves. In order to speed up the execution of these algorithms, our approach relies on the use of extensions to the instruction set architecture. In addition to those specific for cryptography, we use extensions that follow the Single Instruction, Multiple Data (SIMD) parallel computing paradigm. In this model, the processor executes the same operation over a set of data in parallel. We investigated how to apply SIMD to the implementation of elliptic curve algorithms.

As part of our contributions, we design parallel algorithms for prime field and elliptic curve arithmetic. We also design a new three-point ladder algorithm for the scalar multiplication $P + kQ$, and a faster formula for calculating $3P$ on Montgomery curves. These algorithms have found applicability in isogeny-based cryptography. Using SIMD extensions such as SSE, AVX, and AVX2, we develop optimized implementations of the following cryptographic algorithms: X25519, X448, SIDH, ECDH, ECDSA, EdDSA, and qDSA. Performance benchmarks show that these implementations are faster than existing implementations in the state of the art.

Our study confirms that using extensions to the instruction set architecture is an effective tool for optimizing implementations of cryptographic algorithms based on elliptic curves. May this be an incentive not only for those seeking to speed up programs in general but also for computer manufacturers to include more advanced extensions that support the increasing demand for cryptography.

1. Motivation

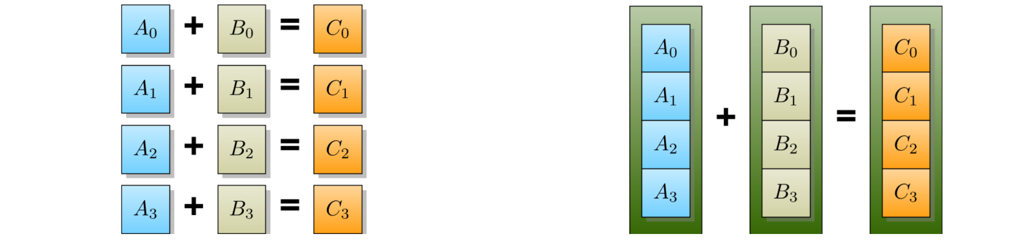
Extensive research efforts have focused on delivering public-key cryptography securely and efficiently. Cryptography based on elliptic curves provides efficient methods due

[*] Summary of a PhD thesis authored by Armando Faz Hernandez and supervised by Julio López. The full text [Faz-Hernandez 2022] is available at https://hdl.handle.net/20.500.12733/6756. Similar summaries appear at CSBC [doi:10.5753/ctd.2023.230156] and CLEIej [doi: 10.19153/cleiej.27.3.3].

to the use of keys shorter than the ones used both in the Rivest-Shamir-Adleman (RSA) cryptosystem [Rivest et al. 1978] and in algorithms based on the Discrete Logarithm Problem (DLP) [ElGamal 1985]. Despite elliptic curve cryptography has been endorsed by international standardization agencies [NIST 2000, ANSI 1998, IEEE 2000] for several years, a recent line of research proposes a *shift to new elliptic curves* with the aim of improving efficiency while preserving high-security guarantees.

With the avalanche of novel elliptic curve proposals, such as the *Montgomery* curves [Montgomery 1987] and the *Edwards* curves [Bernstein et al. 2008], new challenges have appeared. There is still room for improving the algorithms of these alternative curve models. These new algorithms must likely be adapted, or otherwise reformulated considering the upsides and downsides of each model. New optimizations could arise by analyzing the algorithms from the theoretical, computational, and practical standpoints. Therefore, the pathway for designing cryptographic algorithms, their implementation, and their put in practice is currently in progress.

From the computational perspective, a compelling approach for improving performance is using extensions to the instruction set architecture. There exist extensions that support the *Single Instruction, Multiple Data* (SIMD) paradigm characterized in Flynn's taxonomy [Flynn 1966] of parallel computing. In this model, a *vector instruction* encodes an operation that is executed over several data units simultaneously, as shown in Figure 1.



(a) Scalar Processing: Four scalar instructions. **(b) Vector Processing: One vector instruction.**

Figure 1. SIMD vector instructions.

Historically, SIMD processing has been shown effective in the high-performance computing area applied to graphics processing, scientific computing, mathematical simulation, among other domains. In the early days, SIMD execution units were exclusive of large workstations and supercomputers; nowadays, SIMD units are present in commodity processors [Thakkar and Huff 1999, Intel Corporation 2011]. Figure 2 shows the increasing addition of hundreds of new instructions and their applicability to different domains. Lately, instructions now target more specific domains, such as the inclusion of extensions tailored to accelerate cryptography and artificial intelligence algorithms.

2. Research Objectives

Problem Statement. The widespread availability of SIMD execution units in commodity computers, Internet servers, and mobile devices motivates their application to the implementation of cryptographic algorithms. Nonetheless, a few resources explain how to use SIMD units efficiently, and even fewer are dedicated to the case of elliptic curve cryptography, and the secure software development required in this domain. Moreover, it is unclear to what extent these computational resources can help to improve efficiency.

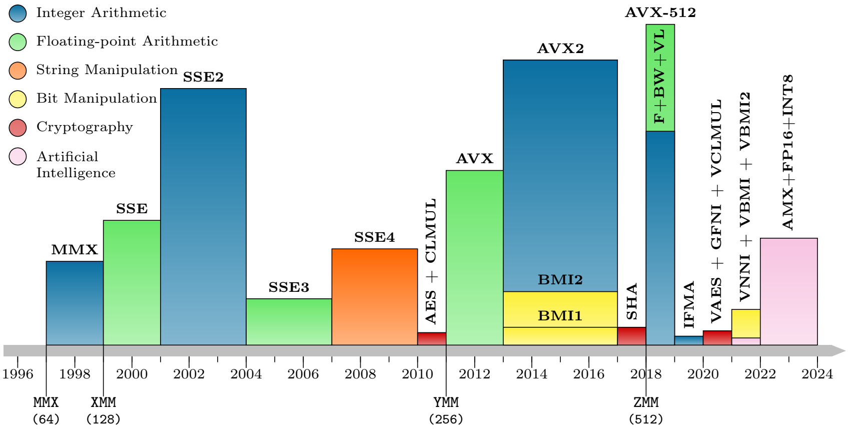


Figure 2. Evolution of SIMD Instructions. Each bar represents an instruction set showing its release date (in the horizontal dimension), the number of instructions (in the vertical dimension), and its domain of application (in the chromatic dimension). The marks show the release date of vector registers.

It is interesting to know how to effectively apply SIMD processing to elliptic curve cryptography. Especially making use of AVX2 and AVX512 [Intel Corporation 2011], the most advanced vector instructions, as well as other extensions found in contemporary computer architectures. For this reason, it is imperative to investigate how to design new algorithms and data structures (or adapt the existing ones) so that implementations of cryptosystems take full advantage of SIMD vector processing.

Thesis Statement. We claim that the execution of algorithms for elliptic curve cryptography can be accelerated through a combination of algorithmic optimizations, implementation techniques, and the use of SIMD processing and other hardware extensions.

Aims. To support this assumption, we investigate algorithmic optimizations and look for implementation techniques for elliptic curve algorithms emphasizing the application of SIMD parallel processing.

An objective of our study is to close the gap between theory and practice. For instance, in addition to proposing parallel algorithms, we also cover their implementation in software. Note that current computer architectures already support hundreds of SIMD instructions such as SSE, AVX, AVX2, and AVX512; and the number of new instructions is increasing in the upcoming computer architectures (as shown in Figure 2). Our research aims to enlighten a pathway for applying SIMD efficiently, to identify some of their limitations, and to show how to apply them to elliptic curve cryptography.

3. Results

Our contributions are the union of several layers of improvements comprising algorithmic optimizations for elliptic curve cryptography, efficient implementation techniques of mathematical field operations using SIMD vector processing, and the immediate applicability of our findings to current information security standards. Now, we briefly describe them.

3.1. Algorithmic Optimizations

For Montgomery curves, we introduced a new *Three-point Ladder Algorithm* that calculates the x-coordinate of $P + kQ$, where P, Q are points on the curve and k is an integer. Our

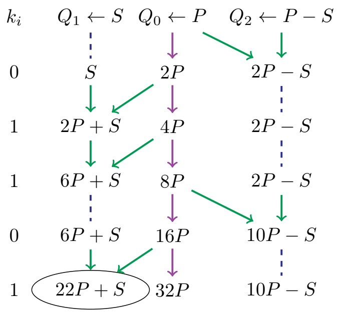


Figure 3. New Three-point Ladder Algorithm. Calculating $22P + S$ from P, S, $P - S$.

algorithm, shown in Figure 3, improves in three aspects. First, it saves a third of the operations required by previously-known algorithms [Costello et al. 2016, Jao et al. 2014]. Second, when P and Q are known in advance, the algorithm runs even faster by means of precomputation. Third, when precomputation is used, fetching precomputed values from memory requires non-secret indexes, which prevents against side-channel attacks.

We show the immediate application of our algorithm to the Diffie-Hellman protocol [Diffie and Hellman 1976], which is the core operation of the Transport Layer Security (TLS) protocol. We apply our algorithm to X25519, X448, qDSA with Montgomery curves, and SIDH/SIKE protocols. The latter is part of Isogeny-based Cryptography, a branch of cryptography looking for algorithms resistant against adversaries with quantum computing power. By using our algorithm, all of these protocols exhibit an enhanced performance. We remark that this improvement is independent of the computer's architecture.

For Montgomery curves, we showed an *optimized formula for tripling points*: given a point P, it calculates $3P$. This operation is relevant for multi-base scalar multiplication methods as well as for the SIDH protocol, which evaluates $3^n P$ for $n > 0$. By applying our formula, we reduce the number of operations by an observable margin. We acknowledge some trade-offs against formulas independently proposed by [Costello and Hisil 2017].

3.2. Implementation Techniques

On the availability of SIMD and other extensions to the instruction set architecture, we speed up implementations of arithmetic operations over prime fields and elliptic curves.

SIMD Implementation of Prime Field Arithmetic. We showed *data structures* and *representation of numbers* suitable for SIMD processing. Our study covers four families of prime moduli corresponding to the ones used in the new elliptic curves. For each family, we show how to perform field operations using scalar and vector instructions. Our benchmark analysis shows that improvements in performance are more significant when operating over larger numbers. On the other hand, when operating on smaller numbers, the vectorized implementation suffers a notorious overhead limiting the amount of improvement.

A better approach to remedy this situation is to take the SIMD's essence to higher abstraction levels. We introduce the notion of n*-way operations*: use the n words of a vector register for calculating n *field operations* in parallel. We follow this approach because it reduces the use of expensive permutation instructions; thus, minimizing the overheads observed in the vectorization for smaller numbers. Armed with n-way operations, we turned our attention to apply them to elliptic curve arithmetic operations.

SIMD Implementation of Elliptic Curve Arithmetic. We apply two-way operations to both the execution of $\mathbb{F}_q$-complete formulas for point addition in Weierstrass curves, and the calculation of the Montgomery ladder step for Montgomery curves. Our implementation strategy consists on using the 256-bit AVX2 unit for simulating two 128-bit units, and each of them can also be seen as two 64-bit units, which are dedicated to field arithmetic.

For Edwards curves, we focused on parallel algorithms for point addition, point doubling, and scalar multiplication. Here, we apply four-way operations to point addition (and doubling), which allows performing the scalar multiplication in parallel. The main criteria of our algorithmic design is to minimize the use of costly permutation instructions. We show that all these strategies speed up the execution of scalar multiplications.

3.3. Optimized Implementation of Cryptographic Algorithms

Building on top the prime field and elliptic curve arithmetic, we developed vectorized implementations of ECDH and ECDSA with the P-384 curve; the X25519, X448, and SIDH protocols; and the EdDSA and qDSA signature schemes. We observed a boost on their performance due to vector processing. Table 1 shows timings of our implementations.

Table 1. Timings of the X25519 and X448 Diffie-Hellman protocols, and EdDSA signature schemes. Entries are 10^3 clock cycles.

Instance	Operation	Haswell	Skylake	Tiger Lake
X25519	Key Generation	43.7	34.5	18.2
	Shared Secret	121.0	99.4	50.7
X448	Key Generation	129.0	107.7	53.7
	Shared Secret	428.1	364.2	168.1
Ed25519	Key Generation	42.8	34.8	18.4
	Signing	48.6	39.5	20.1
	Verification	156.0	123.3	77.1
Ed448	Key Generation	126.7	104.9	54.8
	Signing	132.7	110.1	57.4
	Verification	465.8	409.5	193.6

After profiling EdDSA code, we found its dominant operation is scalar multiplication as shown in Figure 4. So when we speed up this operation, the time of signing decreases by 19% and verification by 24% compared to a previous implementation.

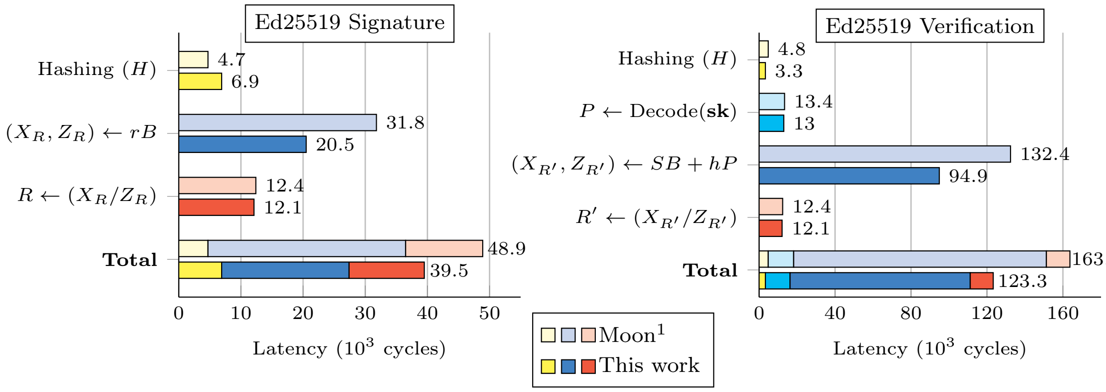


Figure 4. Breakdown of Ed25519's internal operations.

In February 2023, the National Institute of Standards and Technology [NIST 2023] has approved the standardization of EdDSA, which in practical terms means that EdDSA is endorsed to be used on Internet communications massively. This is relevant to secure communication protocols such as TLS, SSH, VPN, and others, which are used globally everyday. Our contributions on accelerating the performance of this algorithm are pertinent.

4. Scientific Production

4.1. Awards

- Honorable Mention Award: *Prêmio Tese Destaque UNICAMP 2022* granted to the best PhD Thesis in Engineering and Technology by the University of Campinas.
- Honorable Mention Award granted to the Best Paper by the SBSeg 2016 committee.
- Finalist at the 36th Thesis and Dissertation Contest organized by the Congress of the Brazilian Computing Society (CSBC 2023).

4.2. Peer-Reviewed Publications

This section includes publications in academic venues that follow a doubly-blinded peer-reviewed process for publication. Table 2 shows venue scores and numbers of citations.

Table 2. Citation counts of peer-reviewed publications (as of June 2024).

Publication	Venue	Venue score		Number of citations			
		Qualis[1]	Estrato[2]	Scopus[3]	WoS[4]	ACM[5]	Google Scholar[6]
[J1]	IEEE Trans. Comp.	A1	A2	62	49	18	119
[J2]	ACM Math. Soft.	A1	A1	32	20	19	45
[C1]	SBSeg	–	A4	–	–	–	3
[C2]	Latincrypt	–	B2	22	15	5	36
[C3]	SAC	–	A3	16	12	–	41
[C4]	SBSeg	–	A4	–	–	–	1
[C5]	SPACE	–	B1	3	1	–	11
[C6]	APKC	–	B3	5	6	4	5
[C7]	SBSeg	–	A4	–	–	–	–
[D1]	SBSeg	–	A4	–	–	–	4

[1] Qualis score corresponds to Period 2017-2020 reported by the Sucupira platform. https://sucupira.capes.gov.br/sucupira/public/index.jsf

[2] Estrato CAPES (Qualis Referência) score reported by the PPGCC/PUCRS University. https://ppgcc.github.io/discentesPPGCC/pt-BR/qualis/

[3] Scopus: https://www.scopus.com/authid/detail.uri?authorId=52363632600

[4] WoS: https://www.webofscience.com/wos/author/record/G-1476-2016

[5] ACM: https://dl.acm.org/profile/81490690977/publications

[6] Google Scholar: https://scholar.google.com/citations?user=XGD6X-EAAAAJ/

Journal Articles

[J1] Faz-Hernández, A., López, J., Ochoa-Jiménez, E., and Rodríguez-Henríquez, F. (2018). A Faster Software Implementation of the Supersingular Isogeny Diffie-Hellman Key Exchange Protocol. IEEE Transactions on Computers. 67(11), 1622–1636. DOI: 10.1109/TC.2017.2771535

[J2] Faz-Hernández, A., López, J., and Dahab, R. (2019). High-performance Implementation of Elliptic Curve Cryptography Using Vector Instructions. ACM Transactions on Mathematical Software, 45(3), 1–35. DOI: 10.1145/3309759

Publications in Conference Proceedings

[C1] Faz-Hernández A. and López J. (2014). On Software Implementation of Arithmetic Operations on Prime Fields using AVX2. In Anais do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2014). 14, 338-341. Sociedade Brasileira de Computação. `DOI: 10.5753/sbseg.2014.20148`

[C2] Faz-Hernández, A., and López, J. (2015). Fast Implementation of Curve25519 Using AVX2. Lecture notes in computer science. Progress in Cryptology – LATINCRYPT 2015. 329-345. Springer. `DOI: 10.1007/978-3-319-22174-8_18`

[C3] Oliveira, T., López, J., Hışıl, H., Faz-Hernández, A., and Rodríguez-Henríquez, F. (2017). How to (Pre-)Compute a Ladder. Lecture notes in computer science. Selected Areas in Cryptography – SAC 2017. 172–191. Springer International Publishing. `DOI: 10.1007/978-3-319-72565-9_9`

[C4] Faz-Hernández A. and López J. (2016). Speeding up Elliptic Curve Cryptography on the P-384 Curve. Anais do XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 170-183. Sociedade Brasileira de Computação. `DOI: 10.5753/sbseg.2016.19306`

[C5] Faz-Hernández, A., Fujii, H., Aranha, D. F., and López, J. (2017). A Secure and Efficient Implementation of the Quotient Digital Signature Algorithm (qDSA). Lecture notes in computer science. Security, Privacy, and Applied Cryptography Engineering. 170–189. Springer. `DOI: 10.1007/978-3-319-71501-8_10`

[C6] Faz-Hernández, A., López, J., and de Oliveira, A. K. D. S. (2018). SoK: A Performance Evaluation of Cryptographic Instruction Sets on Modern Architectures. In APKC '18: Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop. 9-18. ASIA CCS '18: ACM Asia Conference on Computer and Communications Security. `DOI: 10.1145/3197507.3197511`

[C7] Faz-Hernández A. and López J. (2020). Generation of Elliptic Curve Points in Tandem. Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2020). 97-105. Sociedade Brasileira de Computação. `DOI: 10.5753/sbseg.2020.19230`

Dissemination of Science

[D1] Faz-Hernández A. and López J. (2015). Implementação Eficiente e Segura de Algoritmos Criptográficos. Minicursos do XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 93-140. Sociedade Brasileira de Computação. `DOI: 10.5753/sbc.9004.8.3`

[D2] Faz-Hernández, A., López, J., Ochoa-Jiménez, E., and Rodríguez-Henríquez, F. (2018). A Faster Software Implementation of the Supersingular Isogeny Diffie-Hellman Key Exchange Protocol. Revista Avance y Perspectiva. (4)2, Dec, 2018. ISSN: 2448-5977. `URL: https://avanceyperspectiva.cinvestav.mx`

4.3. Software Libraries

We developed a set of software libraries that show implementation techniques and optimizations of the cryptographic algorithms mentioned above. The code uses SIMD processing, and performance benchmarks provide evidence of their superiority. To enable reproducibility, our libraries are released under a permissive software license and are available at an institutional repository: `https://gitlab.ic.unicamp.br/ra142685/phd_libs/`

References

ANSI (1998). Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). Technical Report ANSI X9.62-1998, American National Standards Institute.

Bernstein, D. J., Birkner, P., Joye, M., Lange, T., and Peters, C. (2008). Twisted Edwards Curves. In Vaudenay, S., editor, *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer Berlin Heidelberg.

Costello, C. and Hisil, H. (2017). A Simple and Compact Algorithm for SIDH with Arbitrary Degree Isogenies. In Takagi, T. and Peyrin, T., editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 303–329, Cham. Springer International Publishing.

Costello, C., Longa, P., and Naehrig, M. (2016). Efficient Algorithms for Supersingular Isogeny Diffie-Hellman. In Robshaw, M. and Katz, J., editors, *Advances in Cryptology – CRYPTO 2016*, pages 572–601, Berlin, Heidelberg. Springer Berlin Heidelberg.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

ElGamal, T. (1985). A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. pages 10–18.

Faz-Hernandez, A. (2022). *High-Performance Elliptic Curve Cryptography: A SIMD Approach to Modern Curves*. PhD thesis, University of Campinas, Campinas, Brazil. `https://hdl.handle.net/20.500.12733/6756`.

Flynn, M. (1966). Very high-speed computing systems. *Proceedings of the IEEE*, 54(12):1901–1909.

IEEE (2000). IEEE Standard Specifications for Public-Key Cryptography. `https://doi.org/10.1109/IEEESTD.2000.92292`.

Intel Corporation (2011). Intrinsics for Intel Advanced Vector Extensions. `https://software.intel.com/en-us/isa-extensions`.

Jao, D., De Feo, L., and Plût, J. (2014). Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247.

Montgomery, P. L. (1987). Speeding the Pollard and Elliptic Curve Methods of Factorization. *Mathematics of Computation*, 48(177):243–264.

NIST (2000). Digital Signature Standard (DSS). Technical report, National Institute of Standards and Technology. `http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf`.

NIST (2023). Digital Signature Standard (DSS). Technical Report FIPS PUB 186-5, National Institute of Standards and Technology. `https://doi.org/10.6028/NIST.FIPS.186-5`.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21(2):120–126.

Thakkar, S. and Huff, T. (1999). Internet Streaming SIMD Extensions. *Computer*, 32(12):26–34. `http://doi.org/10.1109/2.809248`.