Elliptic curves in modern cryptography

Andrei Lupsa

April 21, 2025

Abstract

Elliptic curves have a unique set of properties which make their use in cryptography appealing. The abelian group formed by the points of elliptic curves over finite fields provides a basis for the efficiently computable, one-way trapdoor function of point multiplication used in several cryptographic protocols today, ranging from modified Diffie-Hellman key exchange to digital signature algorithms. Despite this, they have provable weaknesses depending on the curve chosen, especially when considering the impacts of post-quantum cryptography. This paper aims to give a solid foundation of the mathematics behind elliptic curve cryptography in order to understand where it falls short.

1 Introduction to elliptic curves

Elliptic curves have a long and beautifully rich history in mathematics. They illustrate perfectly how interconnected mathematics is; elliptic curves have uses ranging across disciplines of mathematics, such as their use in the proof of Fermat's Last Theorem, or the Taniyama—Shimura—Weil conjecture. An interesting application of elliptic curves that is very relevant today is as a trapdoor function in cryptography, but in order to explain their uses it would be useful to cover what they are.

Formally, an elliptic curve is any implicit function where one variable has a degree of 2 and the other has a degree of 3. The kind we are interested in is elliptic curves in **short Weierstrass form**. This means they are in the form $y^2 = x^3 + ax^2 + b$. Examples of elliptic curves in short Weierstrass form can be seen in Figure 1.[1]

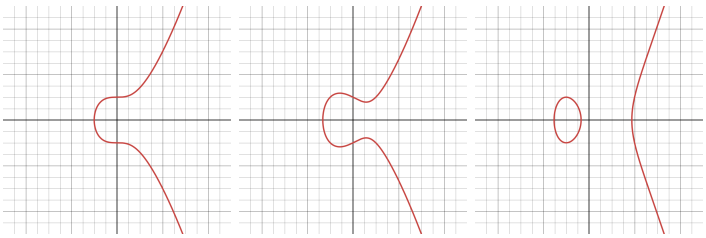

Figure 1: Several elliptic curves with different parameters.

Some important properties of elliptic curves to consider:

1. They are symmetrical across the x-axis.
2. The "tails" of an elliptic curve are asymptotic — they get steeper as you go further along them.

2 Elliptic curve point arithmetic

The points of an elliptic curve have certain properties that enable us to form a group with a specially defined operation of addition. To understand what this means, we need to take a look at abelian groups.

[1] Darrel Hankerson, Alfred J. Menezes and Scott Vanstone, *Guide to Elliptic Curve Cryptography* (Springer Science & Business Media, 2004).

2.1 Abelian groups

An abelian group *under a certain operation* is a group for which that operation is closed, commutative and associative, and for which there exists an inverse for every element and an identity element.[2]

A group G that acts as an abelian group under an operation addition, $+$, is written as $(G, +)$. In other words, an abelian group $(G, +)$ has the following properties for all elements $A \in G$:

1. **Commutativity.** $A + B = B + A$.
2. **Associativity.** $(A + B) + C = A + (B + C)$.
3. **Identity element.** There exists an element I such that $A + I = I + A = A$.
4. **Inverse elements.** There exists an inverse $-A$ for every element A such that $A + -A = -A + A = I$.[3]

An example of an abelian group is the integers under addition $(\mathbb{Z}, +)$, with the identity element 0.

An example of something that is *not* an abelian group is the real numbers under multiplication $(\mathbb{R}, \times)$, because 0 has no inverse. However, the real numbers excluding 0 is an abelian group $(\mathbb{R} \setminus \{0\}, \times)$ with identity element 1.

2.2 Point addition & doubling

The points of an elliptic curve — the pairings (x, y) that satisfy the equation of the curve — form an abelian group under an operation which we define as addition.

Importantly, the addition of two points $(x_1, y_1) + (x_2, y_2) \neq (x_1 + x_2, y_1 + y_2)$; point addition is **not** vector addition. Rather, the addition of points on an elliptic curve is best illustrated graphically:

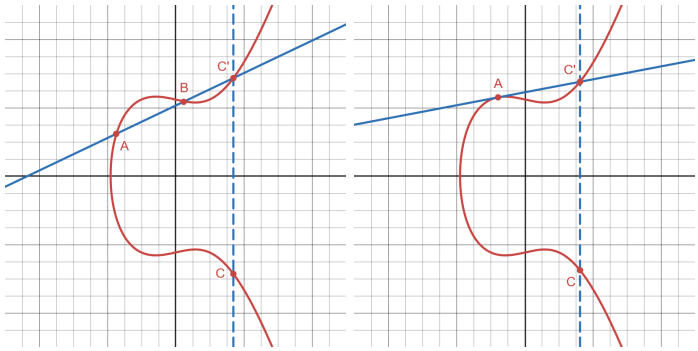

Figure 2: Elliptic curve point addition and doubling..

To add two points A and B:

1. Draw a line between the points AB.
2. A line that intersects an elliptic curve at two points will *always* intersect the curve at a third point (except for special cases, see Section 2.4 on the point at infinity).

   The second property of elliptic curves listed in Section 1 makes it intuitive that even a line that appears "too steep" will always intersect the curve eventually.
3. Take the third point and reflect it across the x-axis to obtain point C.
4. $A + B = C$

The process is similar for adding a point to itself (point doubling), where a line cannot be drawn between two points:

1. Draw a line tangential to the curve at point A.

[2] Jaroslav Ramík, *Pairwise Comparisons Method: Theory and Applications in Decision Making* (Springer Nature, 2020), p. 11.

[3] Hankerson et al., *Guide to Elliptic Curve Cryptography*, p. 11.

2. The line will intersect the curve again at another point.

3. Take the second point and reflect it across the x-axis to obtain point C.

4. $A + A = C$

Also important to note is that every point $A(x, y)$ has an inverse $-A(x, -y)$ in order to satisfy the properties of an abelian group. This property is used when defining the point at infinity (Section 2.4).

2.3 Derivation of point addition & doubling equations

Take two points on an elliptic curve E where $A(x_1, y_1) + B(x_2, y_2) = C(x_3, y_3)$. Let the line through A, B and $-C(x_3, -y_3)$ be L such that

$$E : y^2 = x^3 + ax + b$$
$$L : y = \lambda x + m$$
$$\text{where } \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

The intersection of E and L satisfies the equation

$$x^3 + ax + b = (\lambda x + m)^2$$
$$x^3 + ax + b = \lambda^2 x^2 + 2\lambda x m + m^2$$
$$x^3 - \lambda^2 x^2 + (a - 2\lambda m)x + b - m^2 = 0$$

We also know this equation has roots at x_1, x_2 and x_3, and can be written as

$$(x - x_1)(x - x_2)(x - x_3) = 0$$
$$x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3 = 0$$

Comparing coefficients of x^2 in the two forms of the equation, we can see that

$$\lambda^2 = x_1 + x_2 + x_3$$

and therefore

$$x_3 = \lambda^2 - x_1 - x_2$$

which gives the equation for x_3. To find y_3, we use the fact that λ is also equal to the gradient between $-C(x_3, -y_3)$ and B:

$$\lambda = \frac{-y_3 - y_1}{x_3 - x_1}$$
$$-y_3 - y_1 = \lambda(x_3 - x_1)$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

and we have the equation for y_3.[4]

Adapting these equations for point doubling $A + A = C$ is easy; as the line is no longer through two points but rather the tangent at A, we differentiate E to find the new gradient λ:

$$y^2 = x^3 + ax + b$$
$$2y\frac{dy}{dx} = 3x^2 + a$$
$$\frac{dy}{dx} = \frac{3x^2 + a}{2y}$$

and so

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

Also, as $x_1 = x_2$, the equation for x_3 becomes

$$x_3 = \lambda^2 - 2x_1$$

The equation for y_3 is unchanged.

[4] Prototypeprj, 'Derive equations For point addition & point doubling' (2020).

2.4 Point at infinity

When the points of an elliptic curve are used as an abelian group $(G, +)$, we have to define an identity element. We call this identity element the *point at infinity*, which is written as ∞, $\mathcal{O}$ or 0. This is a point that lies on the curve which we imagine to be at (∞, ∞) and that satisfies the equations

1. $A + \infty = \infty + A = A$
2. $\infty + \infty = \infty$[5]

The point at infinity is the result of two operations:

1. Adding a point to its own inverse, so $A + -A = \infty$.
2. Doubling the point A that is vertically tangential to the curve, where $2A = \infty$. This is because $A = -A$ for that point.

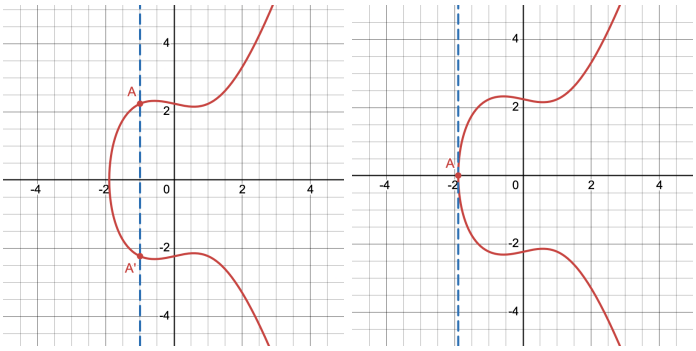

Figure 3: Two cases in which the point at infinity is produced.

2.5 Point multiplication

We also define an operation called point scalar multiplication. Importantly, this isn't multiplying two points by each other: rather, it is an extension of addition (and therefore not a "true" operation) where we repeatedly add a point to itself. $2A = A + A$, $3A = A + A + A$.[6]

Point scalar multiplication is important when considering the elliptic curve discrete logarithm problem (Section 4.1).

3 Elliptic curves over finite fields

Principles of discrete mathematics often emerge when discussing algorithm-related problems, and elliptic curve cryptography is no exception. Two crucial features of ECC are the elliptic curve discrete logarithm problem (ECDLP, Section 4.1) and the efficient one-way calculation of scalar point multiples (Section 2.5). Both of these depend on the use of modular arithmetic to create finite fields.

3.1 Finite fields

Modular arithmetic is often called "clock face arithmetic" because of the likening of restricting calculations modulo n to wrapping numbers around a clock.

Performing arithmetic modulo a number n can be thought of as doing specially defined operations on a finite group — or rather, a special subcategory of group called a field, where multiplication with certain properties is defined as well as addition.

We denote the finite field of integers modulo n as $\mathbb{Z}_n$. This group has only a finite number of elements; for example, the finite field over which we do arithmetic modulo 29 — $\mathbb{Z}_{29}$ — contains only the numbers $\{0, 1, \ldots, 28\}$, which is where the label "finite field" comes from.

[5] Hankerson et al., *Guide to Elliptic Curve Cryptography*
[6] Ibid.

However, not every group produced by modular arithmetic is a finite field. An important axiom in the definition of a field is that in multiplication, every element except the additive identity element (0) has an inverse; i.e. every element A has an element $-A$ for which $A \times -A = 1 \pmod{n}$. This is only true for elements which are co-prime to the modulus n.[7].

Following on from this, we can see that finite fields are only produced modulo a prime p, such that *every* element of the set is co-prime to p except 0, and therefore has an inverse.[8]

3.2 Elliptic curves over finite fields

We can define an elliptic curve $E : y^2 = x^3 + ax + b$ over a field F (written as E/F), where the points of the elliptic curve (x, y) are all such that $x, y \in F$ and $a, b \in F$ (which follows if all coordinates are in F and multiplication is closed).

Elliptic curves can also be defined in this way over a finite field $\mathbb{Z}_n$, where $E : y^2 = x^2 + ax + b \pmod{n}$ — see Figure 4. These curves also have a finite number of points — the number of points, including the point at infinity, is called the *order* of the curve.[9]

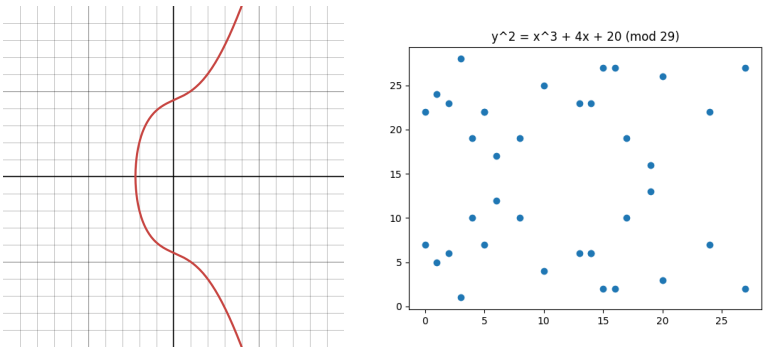

Figure 4: $y^2 = x^3 + 4x + 20$ over $\mathbb{R}$ and $\mathbb{Z}_{29}$

The equations derived in Section 2.3 hold for elliptic curves over finite fields as well, but division modulo p must replace "regular" division over $\mathbb{R}$. This can be done using Euclid's extended algorithm for division[10] or modular exponentiation.

Visually, these adjustments can be explained by allowing the drawn lines to wrap around the field of points until it exactly intersects another point.[11]

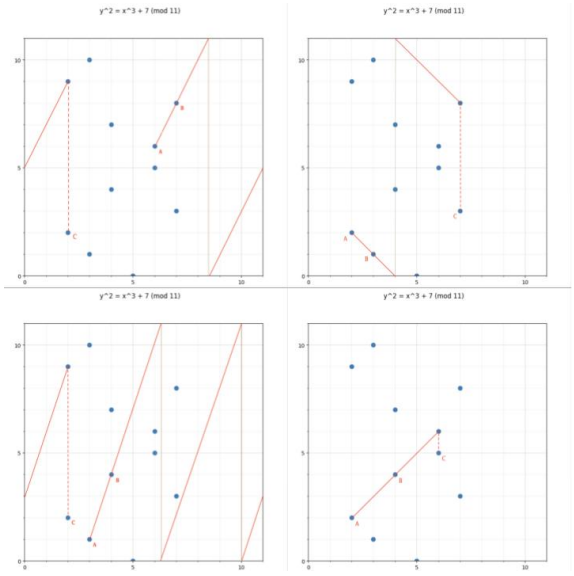

Figure 5: Point multiplication over finite fields. Credit: RareSkills.

[7]László Lovász, József Pelikán and Katalin Vesztergombi, *Discrete Mathematics: Elemenatary and Beyond* (Springer, 2003).
[8]Ibid.
[9]Hankerson et al., *Guide to Elliptic Curve Cryptography*
[10]Lovász et al., *Discrete Maths*
[11]RareSkills, 'Elliptic Curves over Finite Fields' (2023).

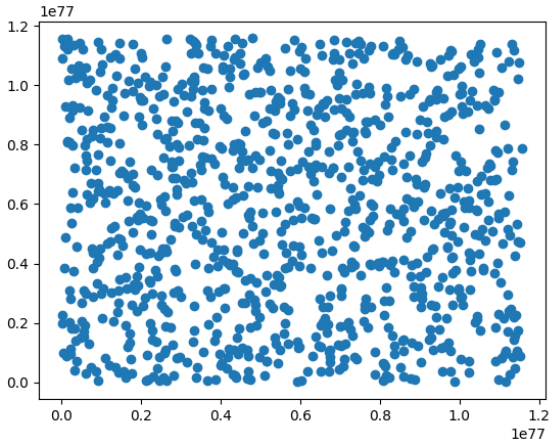

Figure 6: The first 1000 points of the subgroup produced by G on `secp256k1`.

3.3 Cyclical groups & subgroups

An interesting property of elliptic curves over finite fields is that the points of the curve will form a group that is not only abelian, but also cyclical. This means that repeated addition (point multiplication) from a generator point G will produce every point on the curve once, before repeating after reaching the inverse $-G$.

Importantly, this is only true if the field is a prime finite field $\mathbb{Z}_p$. Otherwise, not every point can act as a generator, and some chosen generator points may form a "subgroup" which has a period less than the order of the curve — therefore, not every point can be produced.[12] This is one of the reasons why the curves used in elliptic curve cryptography must be carefully chosen (see Section 5.1).

However, in practise, the moduli chosen for elliptic curves are very large in order to keep the order as large as possible. The parameters for these curves are standardised and given names in order to create a bank of research-proven "strong" curves. One of the curves used for cryptocurrency, named `secp256k1`,[13] has the equation $y^2 = x^3 + 7 \pmod{p}$ where $m = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, with the generator $G($`0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798, 0x483ADA`

It's clear that large numbers are integral to the security of an elliptic curve. A small part of a subgroup of `secp256k1` can be seen in Figure 6, which took an Intel® Core™ i9-7900X CPU @ 3.30GHz approximately 120 seconds to calculate.

4 Public key cryptography with elliptic curves

4.1 The elliptic curve discrete logarithm problem (ECDLP)

One key property of elliptic curve point multiplication is that for certain elliptic curves, attempting to find k where $kG = A$ given G and A is computationally difficult — that is, it is conjectured to be hard to reverse point multiplication.[14]

This property is called the *elliptic curve discrete logarithm problem* due to its similarity to the discrete logarithm problem used in cryptographic protocols like Diffie-Hellman key exchange.[15] The elliptic curve variant is preferable for several reasons, most notably increased speed and efficiency,[16] as well as providing more security for lower key sizes.[17].

Several algorithms exist that speed up the computation of point multiplication, most commonly the sliding window method which decomposes large scalar multiples into several easier-to-calculate operations. However, current algorithms to reverse the calculation are not much better than the naive method (trial-and-error)[18] and have been that way for a long time.

[12] Svetlin Nakov, *Practical Cryptography for Developers* (Svetlin Nakov, 2018).
[13] Certicom, 'SEC 1: Elliptic Curve Cryptography', Technical report (Certicom Research, 2009).
[14] Nakov, *Practical Cryptography for Developers*.
[15] Lovász et al., *Discrete Maths*
[16] NIST, 'Digital Signature Standard (DSS)', Technical report (National Institute of Standards and Technology, 2023).
[17] Dustin Moody et al., 'Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process', Technical report (National Institute of Standards and Technology, 2020).
[18] Hankerson et al., *Guide to Elliptic Curve Cryptography*

This gives elliptic curve cryptography a great profile to be considered for use as a trapdoor function in cryptography.[19] We will cover one protocol as an example of the use of elliptic curves in cryptography, the elliptic curve variant of Diffie-Hellman key exchange.

4.2 Elliptic Curve Diffie-Hellman (ECDH)

One of the most common uses of elliptic curves is in a modified Diffie-Hellman key exchange protocol, which is originally based on number theory's discrete logarithm problem. The set up is as follows:

Two clients, Alice and Bob, choose a generator points G on an elliptic curve $E/\mathbb{Z}_p$ which act as their public keys. Both clients also choose their own private keys p and q respectively, which are random integers between 0 and the order of the elliptic curve.

Alice calculates $p \times G$ to produce the point A, and sends it to Bob. Due to the elliptic curve discrete logarithm problem, any eavesdroppers cannot calculate p given G and A. Bob similarly calculates $q \times G$ to produce B.

Alice then calculates $p \times B$ and Bob calculates $q \times A$ to produce the same point P. This is because

$$\begin{aligned}(G \times p) \times q &= (G \times q) \times p \\ = (G + G + \cdots + G) + (G + G + \cdots + G) \\ &= G \times (p + q)\end{aligned}$$

which follows from the commutative property of elliptic curve point addition.

This way, two clients can communicate over an unsecured network to derive a common secret key using the ECDLP.[20]

5 ECC in the future of cryptography

Despite being robust, efficient and currently in wide-scale usage, the dependence of elliptic curves in cryptography schemes is not future-proof.

5.1 Weaknesses

Programs using elliptic curve cryptography, due to the nature of the algorithms used for point multiplication, are vulnerable to side-channelling attacks.[21] These are attacks which make use of physical properties such as magnetic fluctuations, current and temperature of computers to gain insight into the workings of a program. This is known to even affect virtual machines hosted on the cloud, and the ability of hypervisors to prevent this type of "spying" is limited.[22]

The choice of the curve also potentially opens up any algorithms to several attacks. One of the most dangerous is caused by an elliptic curve with a non-prime modulus and a poorly chosen generator point G; this could inadvertently form a very small cyclical group vulnerable to trial-and-error attack,[23], and such that parameters should be picked from carefully maintained standardised bank of elliptic curves.[24]

Furthermore, elliptic curve schemes are vulnerable to backdoors; `DUAL_EC_DRBG`, a pseudo-random number generator released by NIST together with the NSA, was found to have a specially chosen set of parameters that gave the NSA a backdoor to the algorithm, theoretically enabling them to predict any number the generator would produce.[25] This put every program using `DUAL_EC_DRBG` at risk; not only was the initial curve ambiguously chosen with no formal proof of security, but now anyone who discovered the backdoor could completely crack the random number generator.

[19] Moody et al., 'PQC Status Report, Second Round'

[20] Nakov, *Practical Cryptography for Developers*.

[21] Moody et al., 'PQC Status Report, Second Round'

[22] Shih-Wei Li, John S. Koh and Jason Nieh, *Protecting Cloud Virtual Machines from Hypervisor and Host Operating System Exploits*, in: *28th USENIX Security Symposium (USENIX Security 19)* (Santa Clara, CA: USENIX Association, 2019).

[23] Nakov, *Practical Cryptography for Developers*.

[24] Certicom, 'SEC 1'.

[25] Matthew Green, 'The Many Flaws of Dual_EC_DRBG' (2013).

5.2 Post-quantum cryptography

Arguably the most notable weakness of elliptic curve cryptography is its vulnerability to quantum computers. The ECDLP, much like the regular discrete logarithm problem, is solved in polynomial time by a modified version of Shor's algorithm, with only 1 million Toffoli gates.[26]

This is of particular concern even now because of Store Now, Decrypt Later (SNDL) methodology. Attackers today are constantly intercepting and storing data currently being sent securely over the internet using elliptic curve encryption schemes, in the hope that some day in the future they will have access the quantum computers which can break the encryption used on public internet traffic.[27] This is worrying because some vulnerable data such as confidential government information or personal payment details will still be relevant in the future, when these attackers are able to decrypt data that was previously unreadable. As a result, many bodies are pushing to move away from elliptic curve cryptography in favour of more quantum-safe solutions,[28] like lattice-based cryptography.[29]

6 Conclusion

Elliptic curves brought a novel solution to the cryptography scene; with lower key sizes, efficient algorithms, easy adoption, and perceived hardness, ECC was quickly adopted and is still in use today. The foundation of group theory with the points of a curve over a finite field form a strong mathematical basis for cryptography schemes like ECDH, which made ECC an appealing and secure prospect. However, it is far from foolproof — as evidenced by backdoors and poorly chosen parameters which can jeopardise security — and is especially vulnerable as the era of post-quantum cryptography draws near.

6.1 Accompanying program

This paper was written in conjunction with a Python program, Edu-ECC, which implements the mathematics described above in an ECDH cryptography scheme with several graphical demonstrations. The program, along with documentation, is available on GitHub: https://github.com/shrub719/edu-ecc/.

[26] Daniel Litinski, 'How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates' (2023).
[27] Steve Grobman, 'Quantum Computing's Cyber-Threat to National Security', *PRISM* 9:1 (2020).
[28] Albert Nieto Morales, Arit Kumar Bishwas and Joel Jacob Varghese, 'Quantum-enabled framework for the Advanced Encryption Standard in the post-quantum era' (2025).
[29] Moody et al., 'PQC Status Report, Second Round'

References

Certicom, 'SEC 1: Elliptic Curve Cryptography', Technical report (Certicom Research, 2009).

Green, Matthew, 'The Many Flaws of Dual_EC_DRBG' (2013), https://blog.cryptographyengineering.com/2013/09/18/the-many-flaws-of-dualecdrbg/ (accessed 18 April 2025).

Grobman, Steve, 'Quantum Computing's Cyber-Threat to National Security', *PRISM* 9:1 (2020), pp. 52–67.

Hankerson, Darrel, Menezes, Alfred J. and Vanstone, Scott, *Guide to Elliptic Curve Cryptography* (Springer Science & Business Media, 2004).

Li, Shih-Wei, Koh, John S. and Nieh, Jason, *Protecting Cloud Virtual Machines from Hypervisor and Host Operating System Exploits*, in: *28th USENIX Security Symposium (USENIX Security 19)* (Santa Clara, CA: USENIX Association, 2019), pp. 1357–1374.

Litinski, Daniel, 'How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates' (2023), https://arxiv.org/abs/2306.08585 (accessed 20 April 2025).

Lovász, László, Pelikán, József and Vesztergombi, Katalin, *Discrete Mathematics: Elemenatary and Beyond* (Springer, 2003).

Moody, Dustin et al., 'Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process', Technical report (National Institute of Standards and Technology, 2020).

Morales, Albert Nieto, Bishwas, Arit Kumar and Varghese, Joel Jacob, 'Quantum-enabled framework for the Advanced Encryption Standard in the post-quantum era' (2025).

Nakov, Svetlin, *Practical Cryptography for Developers* (Svetlin Nakov, 2018).

NIST, 'Digital Signature Standard (DSS)', Technical report (National Institute of Standards and Technology, 2023).

Prototypeprj, 'Derive equations For point addition & point doubling' (2020), https://prototypeprj.blogspot.com/2020/07/derive-equations-for-point-addition.html (accessed 11 April 2025).

Ramík, Jaroslav, *Pairwise Comparisons Method: Theory and Applications in Decision Making* (Springer Nature, 2020).

RareSkills, 'Elliptic Curves over Finite Fields' (2023), https://www.rareskills.io/post/elliptic-curves-finite-fields (accessed 19 April 2025).