Elliptic curve cryptography

From Wikipedia, the free encyclopedia

Elliptic curve cryptography (**ECC**) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security.[1]

Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks. Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme. They are also used in several integer factorization algorithms based on elliptic curves that have applications in cryptography, such as Lenstra elliptic curve factorization.

Contents

Rationale

Public-key cryptography is based on the intractability of certain mathematical problems. Early public-key systems are secure assuming that it is difficult to factor a large integer composed of two or more large prime factors. For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible: this is the "elliptic curve discrete logarithm problem" (ECDLP). The security of elliptic curve cryptography depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points. The size of the elliptic curve determines the difficulty of the problem.

The primary benefit promised by elliptic curve cryptography is a smaller key size, reducing storage and transmission requirements, i.e. that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key: for example, a 256-bit elliptic curve public key should provide comparable security to a 3072-bit RSA public key.

The U.S. National Institute of Standards and Technology (NIST) has endorsed elliptic curve cryptography in its Suite B set of recommended algorithms, specifically elliptic curve Diffie–Hellman (ECDH) for key exchange and Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signature. The U.S. National Security Agency (NSA) allows their use for protecting information classified up to top secret with 384-bit keys.[2] However, in August 2015, the NSA announced that it plans to replace Suite B with a new cipher suite due to concerns about quantum computing attacks on ECC.[3]

While the RSA patent expired in 2000, there may be patents in force covering certain aspects of ECC technology. However some argue that the US government elliptic curve digital signature standard (ECDSA; NIST FIPS 186-3) and certain practical ECC-based key exchange schemes (including ECDH) can be implemented without infringing them, including RSA Laboratories[4] and Daniel J. Bernstein[5]

History

The use of elliptic curves in cryptography was suggested independently by Neal Koblitz[6] and Victor S. Miller[7] in 1985. Elliptic curve cryptography algorithms entered wide use in 2004 to 2005.

Theory

For current cryptographic purposes, an *elliptic curve* is a plane curve over a finite field (rather than the real numbers) which consists of the points satisfying the equation

$$y^2 = x^3 + ax + b,$$

along with a distinguished point at infinity, denoted ∞. (The coordinates here are to be chosen from a fixed finite field of characteristic not equal to 2 or 3, or the curve equation will be somewhat more complicated.)

This set together with the group operation of elliptic curves is an Abelian group, with the point at infinity as identity element. The structure of the group is inherited from the divisor group of the underlying algebraic variety.

$$\mathrm{Div}^0(E) \to \mathrm{Pic}^0(E) \simeq E,$$

Cryptographic schemes

Several discrete logarithm-based protocols have been adapted to elliptic curves, replacing the group $(\mathbb{Z}_p)^\times$ with an elliptic curve:

- The elliptic curve Diffie–Hellman (ECDH) key agreement scheme is based on the Diffie–Hellman scheme,
- The Elliptic Curve Integrated Encryption Scheme (ECIES), also known as Elliptic Curve Augmented Encryption Scheme or simply the Elliptic Curve Encryption Scheme,
- The Elliptic Curve Digital Signature Algorithm (ECDSA) is based on the Digital Signature Algorithm,
- The deformation scheme using Harrison's p-adic Manhattan metric,
- The Edwards-curve Digital Signature Algorithm (EdDSA) is based on Schnorr signature and uses twisted Edwards curves,
- The ECMQV key agreement scheme is based on the MQV key agreement scheme,
- The ECQV implicit certificate scheme.

At the RSA Conference 2005, the National Security Agency (NSA) announced Suite B which exclusively uses ECC for digital signature generation and key exchange. The suite is intended to protect both classified and unclassified national security systems and information.[8]

Recently, a large number of cryptographic primitives based on bilinear mappings on various elliptic curve groups, such as the Weil and Tate pairings, have been introduced. Schemes based on these primitives provide efficient identity-based encryption as well as pairing-based signatures, signcryption, key agreement, and proxy re-encryption.

Implementation

Some common implementation considerations include:

Domain parameters

To use ECC, all parties must agree on all the elements defining the elliptic curve, that is, the *domain parameters* of the scheme. The field is defined by *p* in the prime case and the pair of *m* and *f* in the binary case. The elliptic curve is defined by the constants *a* and *b* used in its defining equation. Finally, the cyclic subgroup is defined by its *generator* (a.k.a. *base point*) *G*. For cryptographic application the order of *G*, that is the smallest positive number *n* such that $nG = \infty$, is normally prime. Since *n* is the size of a subgroup of $E(\mathbb{F}_p)$ it follows from Lagrange's theorem that the number $h = \frac{1}{n}|E(\mathbb{F}_p)|$ is an integer. In cryptographic applications this number *h*, called the *cofactor*, must be small ($h \leq 4$) and, preferably, $h = 1$. To summarize: in the prime case, the domain parameters are (p, a, b, G, n, h); in the binary case, they are (m, f, a, b, G, n, h).

Unless there is an assurance that domain parameters were generated by a party trusted with respect to their use, the domain parameters *must* be validated before use.

The generation of domain parameters is not usually done by each participant because this involves computing the number of points on a curve which is time-consuming and troublesome to implement. As a result, several standard bodies published domain parameters of elliptic curves for several common field sizes. Such domain parameters are commonly known as "standard curves" or "named curves"; a named curve can be referenced either by name or by the unique object identifier defined in the standard documents:

- NIST, Recommended Elliptic Curves for Government Use (http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf)
- SECG, SEC 2: Recommended Elliptic Curve Domain Parameters (http://www.secg.org/sec2-v2.pdf)
- ECC Brainpool (RFC 5639), ECC Brainpool Standard Curves and Curve Generation (http://www.ecc-brainpool.org/download/Domain-parameters.pdf)

SECG test vectors are also available.[9] NIST has approved many SECG curves, so there is a significant overlap between the specifications published by NIST and SECG. EC domain parameters may be either specified by value or by name.

If one (despite the above) wants to construct one's own domain parameters, one should select the underlying field and then use one of the following strategies to find a curve with appropriate (i.e., near prime) number of points using one of the following methods:

- Select a random curve and use a general point-counting algorithm, for example, Schoof's algorithm or Schoof–Elkies–Atkin algorithm,
- Select a random curve from a family which allows easy calculation of the number of points (e.g., Koblitz curves), or
- Select the number of points and generate a curve with this number of points using *complex multiplication* technique.[10]

Several classes of curves are weak and should be avoided:

- Curves over $\mathbb{F}_{2^m}$ with non-prime *m* are vulnerable to Weil descent attacks.[11][12]
- Curves such that *n* divides $p^B - 1$ (where *p* is the characteristic of the field – *q* for a prime field, or **2** for a binary field) for sufficiently small *B* are vulnerable to Menezes–Okamoto–Vanstone (MOV) attack[13][14] which applies usual Discrete Logarithm Problem (DLP) in a small degree extension field of $\mathbb{F}_p$ to solve ECDLP. The bound *B* should be chosen so that discrete logarithms in the field $\mathbb{F}_{p^B}$ are at least as difficult to compute as discrete logs on the elliptic curve $E(\mathbb{F}_q)$.[15]

- Curves such that $|E(\mathbb{F}_q)| = q$ are vulnerable to the attack that maps the points on the curve to the additive group of $\mathbb{F}_q$.[16][17][18]

Key sizes

Because all the fastest known algorithms that allow one to solve the ECDLP (baby-step giant-step, Pollard's rho, etc.), need $O(\sqrt{n})$ steps, it follows that the size of the underlying field should be roughly twice the security parameter. For example, for 128-bit security one needs a curve over $\mathbb{F}_q$, where $q \approx 2^{256}$. This can be contrasted with finite-field cryptography (e.g., DSA) which requires[19] 3072-bit public keys and 256-bit private keys, and integer factorization cryptography (e.g., RSA) which requires a 3072-bit value of n, where the private key should be just as large. However the public key may be smaller to accommodate efficient encryption, especially when processing power is limited.

The hardest ECC scheme (publicly) broken to date had a 112-bit key for the prime field case and a 109-bit key for the binary field case. For the prime field case, this was broken in July 2009 using a cluster of over 200 PlayStation 3 game consoles and could have been finished in 3.5 months using this cluster when running continuously.[20] The binary field case was broken in April 2004 using 2600 computers over 17 months.[21]

A current project is aiming at breaking the ECC2K-130 challenge by Certicom, by using a wide range of different hardware: CPUs, GPUs, FPGA.[22]

Projective coordinates

A close examination of the addition rules shows that in order to add two points, one needs not only several additions and multiplications in $\mathbb{F}_q$ but also an inversion operation. The inversion (for given $x \in \mathbb{F}_q$ find $y \in \mathbb{F}_q$ such that $xy = 1$) is one to two orders of magnitude slower[23] than multiplication. Fortunately, points on a curve can be represented in different coordinate systems which do not require an inversion operation to add two points. Several such systems were proposed: in the *projective* system each point is represented by three coordinates (X, Y, Z) using the following relation: $x = \frac{X}{Z}, y = \frac{Y}{Z}$; in the *Jacobian system* a point is also represented with three coordinates (X, Y, Z), but a different relation is used: $x = \frac{X}{Z^2}, y = \frac{Y}{Z^3}$; in the *López–Dahab system* the relation is $x = \frac{X}{Z}, y = \frac{Y}{Z^2}$; in the *modified Jacobian* system the same relations are used but four coordinates are stored and used for calculations (X, Y, Z, aZ^4); and in the *Chudnovsky Jacobian* system five coordinates are used (X, Y, Z, Z^2, Z^3). Note that there may be different naming conventions, for example, IEEE P1363-2000 standard uses "projective coordinates" to refer to what is commonly called Jacobian coordinates. An additional speed-up is possible if mixed coordinates are used.[24]

Fast reduction (NIST curves)

Reduction modulo p (which is needed for addition and multiplication) can be executed much faster if the prime p is a pseudo-Mersenne prime, that is $p \approx 2^d$; for example, $p = 2^{521} - 1$ or $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. Compared to Barrett reduction, there can be an order of magnitude speed-up.[25] The speed-up here is a practical rather than theoretical one, and derives from the fact that the moduli of numbers against numbers near powers of two can be performed efficiently by computers operating on binary numbers with bitwise operations.

The curves over $\mathbb{F}_p$ with pseudo-Mersenne p are recommended by NIST. Yet another advantage of the NIST curves is that they use $a = -3$, which improves addition in Jacobian coordinates.

According to Bernstein and Lange, many of the efficiency-related decisions in NIST FIPS 186-2 are sub-optimal. Other curves are more secure and run just as fast.[26]

Applications

Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

In 1999, NIST recommended 15 elliptic curves. Specifically, FIPS 186-3 has 10 recommended finite fields:

- Five prime fields $\mathbb{F}_p$ for certain primes p of sizes 192, 224, 256, 384, and 521[27] bits. For each of the prime fields, one elliptic curve is recommended.
- Five binary fields $\mathbb{F}_{2^m}$ for m equal 163, 233, 283, 409, and 571. For each of the binary fields, one elliptic curve and one Koblitz curve was selected.

The NIST recommendation thus contains a total of 5 prime curves and 10 binary curves. The curves were ostensibly chosen for optimal security and implementation efficiency.[28]

In 2013, the *New York Times* stated that Dual Elliptic Curve Deterministic Random Bit Generation (or Dual_EC_DRBG) had been included as a NIST national standard due to the influence of NSA, which had included a deliberate weakness in the algorithm and the recommended elliptic curve. RSA Security in September 2013 issued an advisory recommending that its customers discontinue using any software based on Dual_EC_DRBG.[29][30] In the wake of the exposure of Dual_EC_DRBG as "an NSA undercover operation", cryptography experts have also expressed concern over the security of the NIST recommended elliptic curves,[31] suggesting a return to encryption based on non-elliptic-curve groups.

Security

Side-channel attacks

Unlike most other DLP systems (where it is possible to use the same procedure for squaring and multiplication), the EC addition is significantly different for doubling ($P = Q$) and general addition ($P \neq Q$) depending on the coordinate system used. Consequently, it is important to counteract side channel attacks (e.g., timing or simple/differential power analysis attacks) using, for example, fixed pattern window (a.k.a. comb) methods[32] (note that this does not increase computation time). Alternatively one can use an Edwards curve; this is a special family of elliptic curves for which doubling and addition can be done with the same operation.[33] Another concern for ECC-systems is the danger of fault attacks, especially when running on smart cards.[34]

Backdoors

Cryptographic experts have expressed concerns that the National Security Agency has inserted a kleptographic backdoor into at least one elliptic curve-based pseudo random generator.[35] Internal memos leaked by former NSA contractor, Edward Snowden, suggest that the NSA put a backdoor in the Dual_EC_DRBG standard.[36] One analysis of the possible backdoor concluded that an adversary in possession of the algorithm's secret key could obtain encryption keys given only 32 bytes of ciphertext.[37]

The SafeCurves project has been launched in order to catalog curves that are easy to securely implement and are designed in a fully publicly verifiable way to minimize the chance of a backdoor.[38]

Quantum computing attacks

In contrast with its current standing over RSA, elliptic curve cryptography is expected to be more vulnerable to an attack based on Shor's algorithm.[39] In theory, making a practical attack feasible many years before an attack on an equivalently secure RSA scheme is possible.[40] This is because smaller elliptic curve keys are needed to match the classical security of RSA. The work of Proos and Zalka show how a quantum computer for breaking 2048-bit RSA requires roughly 4096 qubits, while a quantum computer to break the equivalently secure 224-bit Elliptic Curve Cryptography requires between 1300 and 1600 qubits.

To avoid quantum computing concerns, an elliptic curve-based alternative to Elliptic Curve Diffie Hellman which is not susceptible to Shor's attack is the Supersingular Isogeny Diffie–Hellman Key Exchange of De Feo, Jao and Plut. It uses elliptic curve isogenies to create a drop-in replacement for the quantum attackable Diffie–Hellman and Elliptic curve Diffie–Hellman key exchanges. This key exchange uses the same elliptic curve computational primitives of existing elliptic curve cryptography and requires computational and transmission overhead similar to many currently used public key systems.[41]

In August, 2015, NSA announced that it planned to transition "in the not distant future" to a new cipher suite that is resistant to quantum attacks. "Unfortunately, the growth of elliptic curve use has bumped up against the fact of continued progress in the research on quantum computing, necessitating a re-evaluation of our cryptographic strategy."[3]

Patents

At least one ECC scheme (ECMQV) and some implementation techniques are covered by patents.

Alternative representations

Alternative representations of elliptic curves include:

- Hessian curves
- Edwards curves
- Twisted curves
- Twisted Hessian curves
- Twisted Edwards curve
- Doubling-oriented Doche–Icart–Kohel curve
- Tripling-oriented Doche–Icart–Kohel curve
- Jacobian curve
- Montgomery curve

See also

- Cryptocurrency
- Curve25519
- DNSCurve
- ECC patents
- ECDH
- Elliptic Curve Digital Signature Algorithm
- ECMQV
- Elliptic curve point multiplication
- Homomorphic Signatures for Network Coding
- Pairing-based cryptography
- Public-key cryptography
- Quantum cryptography

Notes

1. Commercial National Security Algorithm Suite and Quantum Computing FAQ (https://cryptome.org/2016/01/CNSA-Suite-and-Quantum-Computing-FAQ.pdf) U.S. National Security Agency, January 2016

2. "Fact Sheet NSA Suite B Cryptography" (https://web.archive.org/web/20090207005135/http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml). *U.S. National Security Agency*. Archived from the original (http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml) on 2009-02-07.
3. https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml
4. RSA Laboratories. "6.3.4 Are elliptic curve cryptosystems patented?" (http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/are-elliptic-curve-cryptosystems-patented.htm). Retrieved 15 December 2014.
5. Bernstein, D. J. "Irrelevant patents on elliptic-curve cryptography" (http://cr.yp.to/ecdh/patents.html).
6. Koblitz, N. (1987). "Elliptic curve cryptosystems". *Mathematics of Computation*. **48** (177): 203–209. JSTOR 2007884 (https://www.jstor.org/stable/2007884). doi:10.2307/2007884 (https://doi.org/10.2307%2F2007884).
7. Miller, V. (1985). "Use of elliptic curves in cryptography". *CRYPTO*. Lecture Notes in Computer Science. **85**: 417–426. ISBN 978-3-540-16463-0. doi:10.1007/3-540-39799-X_31 (https://doi.org/10.1007%2F3-540-39799-X_31).
8. "The Case for Elliptic Curve Cryptography" (https://web.archive.org/web/20090117023500/http://www.nsa.gov/business/programs/elliptic_curve.shtml). *NSA*. Archived from the original (http://www.nsa.gov/business/programs/elliptic_curve.shtml) on 2009-01-17.
9. http://www.secg.org/download/aid-390/gec2.pdf
10. Lay, G.; Zimmer, H. (1994). "Constructing elliptic curves with given group order over large finite fields". *Algorithmic Number Theory Symposium*. Lecture Notes in Computer Science. **877**: 250–263. doi:10.1007/3-540-58691-1_64 (https://doi.org/10.1007%2F3-540-58691-1_64).
11. Galbraith, S. D.; Smart, N. P. (1999). "A cryptographic application of the Weil descent". *Cryptography and Coding*. Lecture Notes in Computer Science. **1746**: 799. doi:10.1007/3-540-46665-7_23 (https://doi.org/10.1007%2F3-540-46665-7_23).
12. Gaudry, P.; Hess, F.; Smart, N. P. (2000). "Constructive and destructive facets of Weil descent on elliptic curves" (http://www.hpl.hp.com/techreports/2000/HPL-2000-10.pdf) (PDF). *Hewlett Packard Laboratories Technical Report*.
13. Menezes, A.; Okamoto, T.; Vanstone, S. A. (1993). "Reducing elliptic curve logarithms to logarithms in a finite field". *IEEE Transactions on Information Theory*. **39**: 1639–1646. doi:10.1109/18.259647 (https://doi.org/10.1109%2F18.259647).
14. Hitt, L. (2006). "On an Improved Definition of Embedding Degree" (http://eprint.iacr.org/2006/415). *IACR ePrint report*. **415**.
15. IEEE P1363 (http://grouper.ieee.org/groups/1363/P1363/index.html), section A.12.1
16. Semaev, I. (1998). "Evaluation of discrete logarithm in a group of *p*-torsion points of an elliptic curve in characteristic *p*". *Mathematics of Computation*. **67** (221): 353–356. doi:10.1090/S0025-5718-98-00887-4 (https://doi.org/10.1090%2FS0025-5718-98-00887-4).
17. Smart, N. (1999). "The discrete logarithm problem on elliptic curves of trace one". *Journal of Cryptology*. **12** (3): 193–196. doi:10.1007/s001459900052 (https://doi.org/10.1007%2Fs001459900052).
18. Satoh, T.; Araki, K. (1998). "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves". *Commentarii Mathematici Universitatis Sancti Pauli*. **47**.
19. NIST, Recommendation for Key Management—Part 1: general (http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf), Special Publication 800-57, August 2005.
20. http://lacal.epfl.ch/page81774.html
21. "Certicom Announces Elliptic Curve Cryptography Challenge Winner" (https://web.archive.org/web/20110719233751/https://www.certicom.com/index.php/2004-press-releases/36-2004-press-releases/300-solution-required-team-of-mathematicians-2600-computers-and-17-months-). *Certicom*. April 27, 2004. Archived from the original (http://www.certicom.com/index.php/2004-press-releases/36-2004-press-releases/300-solution-required-team-of-mathematicians-2600-computers-and-17-months-) on 2011-07-19.
22. http://www.ecc-challenge.info/
23. Hitchcock, Y.; Dawson, E.; Clark, A.; Montague, P. (2002). "Implementing an efficient elliptic curve cryptosystem over GF(p) on a smart card" (https://web.archive.org/web/20060327202009/http://anziamj.austms.org.au/V44/CTAC2001/Hitc/Hitc.pdf) (PDF). *ANZIAM Journal*. **44**. Archived from the original (http://anziamj.austms.org.au/V44/CTAC2001/Hitc/Hitc.pdf) (PDF) on 2006-03-27.
24. Cohen, H.; Miyaji, A.; Ono, T. (1998). "Efficient Elliptic Curve Exponentiation Using Mixed Coordinates". *Advances in Cryptology – AsiaCrypt '98*. Lecture Notes in Computer Science. **1514**: 51–65. ISBN 978-3-540-65109-3. doi:10.1007/3-540-49649-1_6 (https://doi.org/10.1007%2F3-540-49649-1_6).
25. Brown, M.; Hankerson, D.; Lopez, J.; Menezes, A. (2001). "Software Implementation of the NIST Elliptic Curves Over Prime Fields". *Topics in Cryptology – CT-RSA 2001*. Lecture Notes in Computer Science. **2020**: 250–265. ISBN 978-3-540-41898-6. doi:10.1007/3-540-45353-9_19 (https://doi.org/10.1007%2F3-540-45353-9_19).
26. Daniel J. Bernstein & Tanja Lange. "SafeCurves: choosing safe curves for elliptic-curve cryptography" (https://safecurves.cr.yp.to/). Retrieved 1 December 2013.
27. The sequence may seem suggestive of a typographic error. Nevertheless, the last value is 521 and not 512 bits.
28. FIPS PUB 186-3, Digital Signature Standard (DSS) (http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf).
29. Kim Zetter, RSA Tells Its Developer Customers: Stop Using NSA-Linked Algorithm (https://www.wired.com/threatlevel/2013/09/rsa-advisory-nsa-algorithm/) *Wired*, 19 September 2013. "Recommending against the use of SP 800-90A Dual Elliptic Curve Deterministic Random Bit Generation: NIST strongly recommends that, pending the resolution of the security concerns and the re-issuance of SP 800-90A, the Dual_EC_DRBG, as specified in the January 2012 version of SP 800-90A, no longer be used."
30. "Due to the debate around the Dual EC DRBG standard highlighted recently by the National Institute of Standards and Technology (NIST), NIST re-opened for public comment its SP 800-90 standard which covers Pseudo-random Number Generators (PRNG)." csrc.nist.gov (http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-90-A%20Rev%201%20B%20and%20C)
31. Bruce Schneier (5 September) "I no longer trust the constants. I believe the NSA has manipulated them through their relationships with industry." See Are the NIST Standard Elliptic Curves Back-doored? (http://it.slashdot.org/firehose.pl?op=view&type=story&sid=13/09/11/1224252), *Slashdot*, 11 September 2013.
32. Hedabou, M.; Pinel, P.; Beneteau, L. (2004). "A comb method to render ECC resistant against Side Channel Attacks" (http://eprint.iacr.org/2004/342.pdf) (PDF).
33. http://blog.cr.yp.to/20140323-ecdsa.html
34. See, for example, Biehl, Ingrid; Meyer, Bernd; Müller, Volker (2000). "Differential Fault Attacks on Elliptic Curve Cryptosystems". *Advances in Cryptology – CRYPTO 2000*. Lecture Notes in Computer Science. **1880**: 131–146. ISBN 978-3-540-67907-3. doi:10.1007/3-540-44598-6_8 (https://doi.org/10.1007%2F3-540-44598-6_8).
35. https://www.schneier.com/essay-198.html
36. "Government Announces Steps to Restore Confidence on Encryption Standards" (http://bits.blogs.nytimes.com/2013/09/10/government-announces-steps-to-restore-confidence-on-encryption-standards/). *NY Times - Bits Blog*. Retrieved 2015-11-06.
37. http://rump2007.cr.yp.to/15-shumow.pdf
38. Bernstein, Daniel J.; Lange, Tanja. "SafeCurves: choosing safe curves for elliptic-curve cryptography" (http://safecurves.cr.yp.to/). Retrieved October 1, 2016.
39. Michael A. Nielsen; Isaac L. Chuang (9 December 2010). *Quantum Computation and Quantum Information:* (https://books.google.com/books?id=-s4DEy7o-a0C&pg=PA202) (10th Anniversary ed.). Cambridge University Press. pp. 202–. ISBN 978-1-139-49548-6.
40. Proos, John; Zalka. "Shor's discrete logarithm quantum algorithm for elliptic curves". QIC. arXiv:quantph/0301141 (https://arxiv.org/abs/quantph/0301141).
41. De Feo, Luca; Jao, Plut. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies" (https://eprint.iacr.org). *Cryptology ePrint Archive, Report 2011/506*. IACR. Archived from the original (https://eprint.iacr.org/2011/506) on 2011. Retrieved 3 May 2014.

References

- Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography (http://www.secg.org/sec1-v2.pdf), Version 1.0, September 20, 2000. (archived (https://web.archive.org/web/20141111191126/http://www.secg.org/sec1-v2.pdf) as if Nov 11, 2014)
- D. Hankerson, A. Menezes, and S.A. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag, 2004.
- I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society 265, Cambridge University Press, 1999.
- I. Blake, G. Seroussi, and N. Smart, editors, *Advances in Elliptic Curve Cryptography*, London Mathematical Society 317, Cambridge University Press, 2005.
- L. Washington, *Elliptic Curves: Number Theory and Cryptography*, Chapman & Hall / CRC, 2003.
- The Case for Elliptic Curve Cryptography (https://web.archive.org/web/20090117023500/http://www.nsa.gov/business/programs/elliptic_curve.shtml), National Security Agency (archived January 17, 2009)
- Online Elliptic Curve Cryptography Tutorial (http://www.certicom.com/index.php/ecc-tutorial), Certicom Corp. (archived here (https://web.archive.org/web/20160309033943/http://certicom.com/index.php/ecc-tutorial) as of March 3, 2016)
- K. Malhotra, S. Gardner, and R. Patz, Implementation of Elliptic-Curve Cryptography on Mobile Healthcare Devices, Networking, Sensing and Control, 2007 IEEE International Conference on, London, 15–17 April 2007 Page(s):239–244
- Saikat Basu, A New Parallel Window-Based Implementation of the Elliptic Curve Point Multiplication in Multi-Core Architectures (http://ijns.jalaxy.com.tw/contents/ijns-v14-n2/ijns-2012-v14-n2-p101-108.pdf), International Journal of Network Security, Vol. 13, No. 3, 2011, Page(s):234–241 (archived here (https://web.archive.org/web/20160304121101/http://ijns.jalaxy.com.tw/contents/ijns-v14-n2/ijns-2012-v14-n2-p101-108.pdf) as of March 4, 2016)

- Christof Paar, Jan Pelzl, "Elliptic Curve Cryptosystems" (http://wiki.crypto.rub.de/Buch/movies.php), Chapter 9 of "Understanding Cryptography, A Textbook for Students and Practitioners". (companion web site contains online cryptography course that covers elliptic curve cryptography), Springer, 2009. (archived here (https://archive.is/20121208212741/http://wiki.crypto.rub.de/Buch/movies.php) as of April 20, 2016)
- Luca De Feo, David Jao, Jerome Plut, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies (http://eprint.iacr.org/2011/506), Springer 2011. (archived here (https://web.archive.org/web/20120507200407/http://eprint.iacr.org/2011/506) as of May 7, 2012)

- Jacques Vélu, *Courbes elliptiques (...)*, Société Mathématique de France, **57**, 1-152, Paris, 1978. (http://archive.numdam.org/ARCHIVE/MSMF/MSMF_1978__57_/MSMF_1978__57__1_0/MSMF_1978__57__1_0.pdf)

External links

Wikimedia Commons has media related to ***Elliptic curve***.

- Certicom ECC Tutorial (https://www.certicom.com/content/certicom/en/ecc-tutorial.html)
- a relatively easy to understand primer on elliptic curve cryptography (https://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/)
- Elliptic curves and their implementation (http://www.imperialviolet.org/2010/12/04/ecc.html) by Adam Langley (OpenSSL dev).
- Interactive introduction to elliptic curves and elliptic curve cryptography with Sage (https://web.archive.org/web/20120301091325/http://sagenb.org/home/pub/1126/) by Maike Massierer (http://www.maths.unsw.edu.au/~maikemassierer/) and the CrypTool (https://www.cryptool.org/en/) team
- How did the NSA hack our e-mails? (http://www.numberphile.com/videos/NSA1.html) explained by Mathematician Edward Frenkel
- How to design an elliptic-curve signature system (http://blog.cr.yp.to/20140323-ecdsa.html) by Daniel J. Bernstein

Categories: Elliptic curve cryptography | Public-key cryptography | Finite fields

---