Cryptography

Elliptic Curve Cryptography

Prof. Dr. Heiko Knospe

TH Köln – University of Applied Sciences

April 23, 2025

Groups for Cryptographic Applications

The multiplicative group $\mathbb{Z}_p^*$ of integers modulo a prime number p is used in many cryptographic algorithms, for example in the Diffie-Hellman key exchange. One assumes that the *discrete logarithm problem* is hard in $\mathbb{Z}_p^*$.

A well-proven alternative is the additive group of points on an *elliptic curve* over a finite field. Elliptic curves are defined by a (short) *Weierstrass equation*

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

over a field K. In this chapter, we assume that $2 \neq 0$ and $3 \neq 0$ in K.

Points on the curve have two affine coordinates. We also look at the corresponding *projective curve* with *three projective coordinates*.

Points in Affine and Projective Spaces

Points in the two-dimensional *affine space* over K have two coordinates $(x, y) \in K^2$. Points in the two-dimensional *projective space* $\mathbb{P}^2(K)$ have *three* coordinates

$$[x : y : z]$$

with $[x : y : z] \neq [0 : 0 : 0]$. Elements in $\mathbb{P}^2(K)$ correspond to *lines* in K^3. The points $[x : y : z]$ and $[\lambda x : \lambda y : \lambda z]$ are identified for $\lambda \in K^*$.

Example: $[2 : 3 : 1] = [4 : 6 : 2]$, but $[1 : 1 : 1] \neq [2 : 2 : 1]$.

Affine and projective spaces exist for any dimension, but for our purposes we only look at dimension 2.

Embedding of the Affine Space in the Projective Spaces

Most (but not all!) projective points can be transformed into the form $[x : y : 1]$, and therefore correspond to an affine point (x, y).

We can map $(x, y) \in K^2$ to $[x : y : 1] \in \mathbb{P}^2(K)$. This gives an injective map and the image consists of all projective points $[x : y : z]$ with $z \neq 0$, since $[x : y : z] = [\frac{x}{z} : \frac{y}{z} : 1]$.

However, the projective space has extra points of the form $[x : y : 0]$, where $x \neq 0$ or $y \neq 0$ and the last coordinate is zero. These can be viewed as *points at infinity*, corresponding to lines that are parallel to the (x, y) plane.

Weierstrass Equation

Definition

Let K be a field and $a, b \in K$, then the short Weierstrass equation

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

defines an affine curve and a set of points in K^2.

We replace x by $\frac{x}{z}$ and y by $\frac{y}{z}$, multiply the equation by z^3 and obtain the *projective curve*

$$y^2 z = x^3 + axz^2 + bz^3.$$

Points $[x : y : 1] \in \mathbb{P}^2(K)$ on the projective curve correspond to points $(x, y) \in K^2$ on the affine curve. However, the projective curve has one additional point $O = [0 : 1 : 0]$ *at infinity*.

Elliptic Curves

Let $y^2 = x^3 + ax + b$ be a Weierstrass curve over a field K. Then $\Delta = -16(4a^3 + 27b^2)$ is called the *discriminant* of the curve. If Δ is nonzero in K, then the curve is *nonsingular*.

Definition

An *elliptic curve* E over a field K is a nonsingular projective curve defined by a Weierstrass equation. The set of points on E with coordinates in K is denoted by $E(K)$:

$$\begin{aligned} E(K) &= \{[x : y : z] \in \mathbb{P}^2(K) \mid y^2 z = x^3 + axz^2 + bz^3\} \\ &= \{(x, y) \in K \times K \mid y^2 = x^3 + ax + b\} \cup \{[0 : 1 : 0]\}. \end{aligned}$$

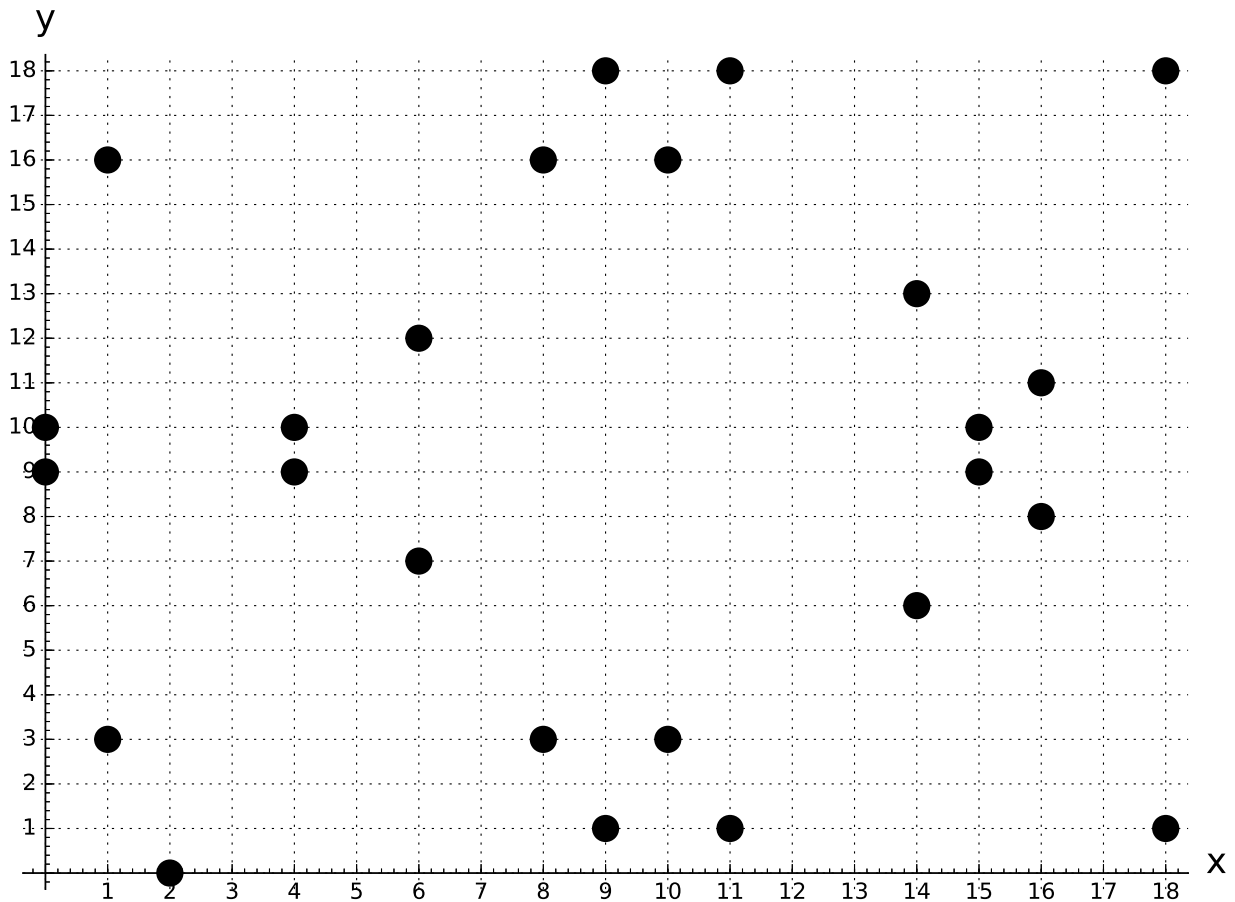
Example: Elliptic Curve over a Finite Field

Consider the elliptic curve $E : y^2 = x^3 + 3x + 5$ over $K = GF(19)$. We can check points $(x, y) \in GF(19)^2$ and find that the following points are lying on the elliptic curve E:

```
sage:   E=EllipticCurve(GF(19),[3,5])
sage:   E.points()
[(0:1:0), (0:9:1), (0:10:1), (1:3:1), (1:16:1),
 (2:0:1), (4: 9:1), (4:10:1), (6:7:1), (6:12:1),
 (8:3:1), (8:16:1), (9:1:1), (9:18:1), (10:3:1),
 (10:16:1), (11:1:1),  (11:18:1), (14:6:1),
 (14:13:1), (15:9:1), (15:10:1), (16:8:1),
 (16:11:1), (18:1:1), (18:18:1)]
```

The points on the affine curve are of the form $[x : y : 1]$, and there is one extra point $O = [0 : 1 : 0]$ at infinity. $E(K)$ contains 26 points.

Example: Points of an Elliptic Curve over a Finite Field



Points on the affine curve $y^2 = x^3 + 3x + 5$ *over* $GF(19)$.

Group Law

A very important fact is that points in $E(K)$ *can be added*. However, this is **not** the usual vector addition in K^2.

The identity element is the point O at infinity. A line through two points P and Q (or the tangent if $P = Q$) intersects the elliptic curve at a third point R and we set $P + Q + R = O$, i.e., $P + Q = -R$, where $-R = -(x, y) = (x, -y)$ is the reflected point.

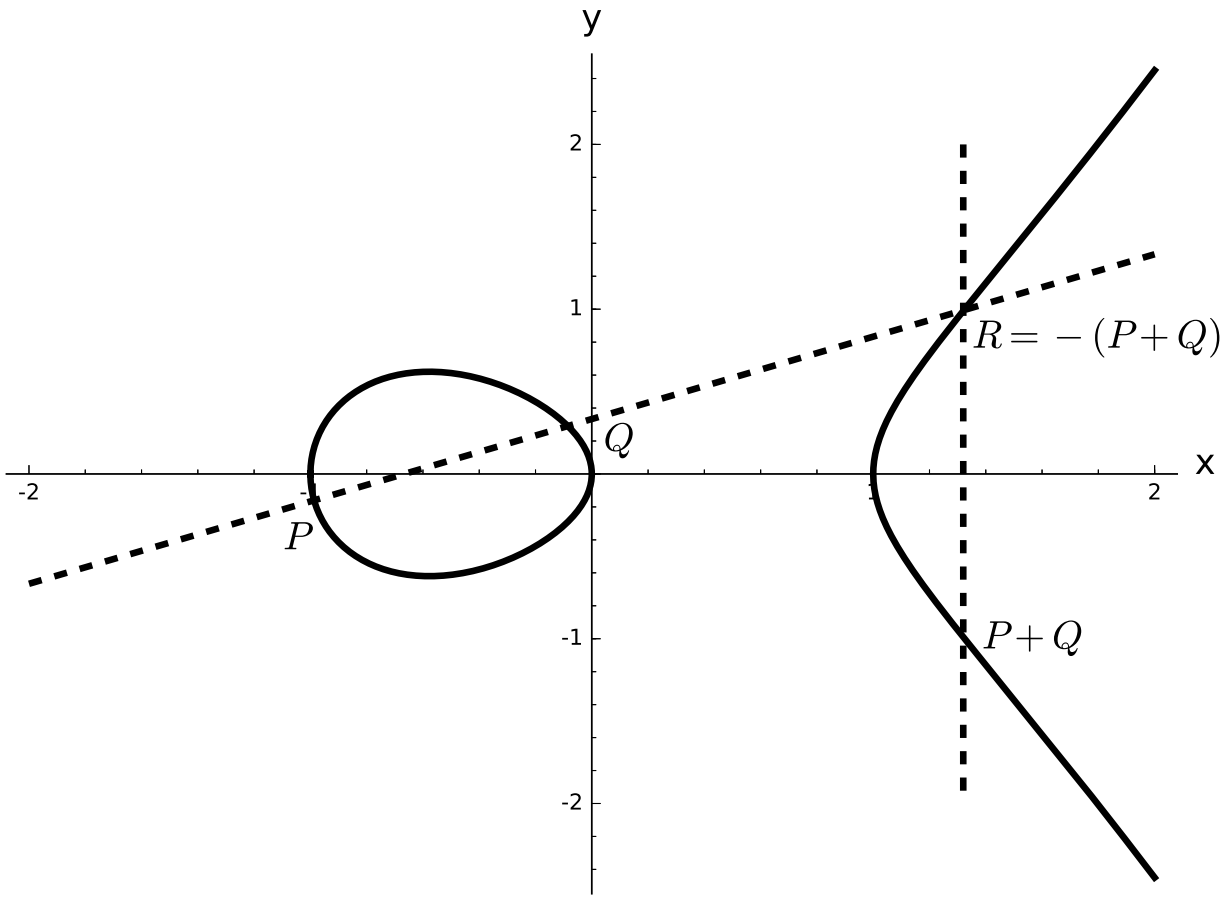
Note: Since E is defined by a cubic equation, two solutions in K (or a double zero) yield a third solution in K.

Theorem

$E(K)$ forms an abelian group with identity element O.

Example: Consider the elliptic curve $E : y^2 = x^3 + 3x + 5$ over $K = GF(19)$ (see above). We have $E(K) \cong \mathbb{Z}_{26} \cong \mathbb{Z}_{13} \times \mathbb{Z}_2$.

Addition of Points



The line through P and Q intersects the curve in R and one has $P + Q = -R$. Note: the picture shows points with real coordinates.

Formulas for Point Addition and Doubling

Let E be an elliptic curve over K, defined by the Weierstrass equation $y^2 = x^3 + ax + b$. Let $P = (x_1, y_1)$, $Q = (x_2, y_2) \in E(K)$, $P \neq O$, $Q \neq O$ and $P \neq -Q$. Then:

$$P + Q = (x_3, y_3),\ x_3 = m^2 - x_1 - x_2,\ y_3 = m(x_1 - x_3) - y_1,$$

$$\text{where } m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q. \end{cases}$$

m is the slope of the line through P and Q or the tangent line, if $P = Q$. Furthermore,

$$-P = (x_1, -y_1) \quad \text{if } P \neq O.$$

Fast Multiplication of Points

The algorithms in elliptic curve cryptography require a fast *multiplication* of points. Analogous to the square-and-multiply algorithm for a fast exponentiation, multiplying a point by a factor can be done recursively by doubling and adding points (*double-and-add algorithm*).

Example: $17 \cdot P = 2 \cdot (2 \cdot (2 \cdot (2 \cdot P))) + P$.

Cryptographic Applications

For cryptographic use, one chooses a finite field $K = GF(p)$, where p is a prime (or $K = GF(2^m)$ where $m \in \mathbb{N}$), an elliptic curve E over K and a *base point* $g \in E(K)$.

The point g generates a cyclic subgroup

$$G = < g > \subset E(K)$$

of order $n = ord(g)$. The order should be a large prime number or at least contain a large prime factor. The cofactor is defined as $h = \frac{ord(E(K))}{n}$. Usually, the *domain parameters* are chosen such that h is small or equal to 1.

Number of Points on an Elliptic Curve

There are efficient algorithms to compute the order of $E(K)$ for a finite field K. Hasse's Theorem gives the *approximate number of points*:

Theorem

Let E be an elliptic curve over $K = GF(p)$. Then

$$| p + 1 - ord(E(GF(p))) | \leq 2\sqrt{p}.$$

Hence $E(GF(p))$ has about the same size as p since $2\sqrt{p}$ is much smaller than p.

Note: the Theorem also holds over $GF(q)$, where $q = p^n$.

Discrete Logarithm

Definition

Let E be an elliptic curve over a finite field K, $g \in E(K)$, $G = < g >$, $n = ord(G)$ and $A \in G$. Then the unique integer $0 \leq a < n$ such that

$$a \cdot g = A$$

is called the *discrete logarithm* $\log_g(A)$ of A.

The security of elliptic curve cryptography relies on the hardness of the discrete logarithm (DL) problem in the group $G \subset E(K)$. The elliptic curve and its so-called *domain parameters* must be carefully chosen since there are less secure curves, where the computation of discrete logarithms can be reduced to an easier DL problem.

Example

Elliptic curve cryptography (ECC) is now widely standardized by national and international organizations (e.g., ISO, ANSI, NIST, IEEE, IETF), and usually one of the proposed curves is chosen.

Consider the curve `brainpoolP256r1` (RFC 5639). The curve is defined by the Weierstrass equation $y^2 = x^3 + ax + b$ over a 256-bit field $K = GF(p)$. The base point $g = (x_g, y_g)$ generates the full group $G = E(K)$ and $n = ord(g)$ is a 256-bit prime number.

```
p =  A9FB57DBA1EEA9BC3E660A909D838D726E3BF623D52620282013481D1F6E5377
a =  7D5A0975FC2C3057EEF67530417AFFE7FB8055C126DC5C6CE94A4B44F330B5D9
b =  26DC5C6CE94A4B44F330B5D9BBD77CBF958416295CF7E1CE6BCCDC18FF8C07B6
g =  (xg,yg)
xg=  8BD2AEB9CB7E57CB2C4B482FFC81B7AFB9DE27E1E3BD23C23A4453BD9ACE3262
yg=  547EF835C3DAC4FD97F8461A14611DC9C27745132DED8E545C1D54C72F046997
n  = A9FB57DBA1EEA9BC3E660A909D838D718C397AA3B561A6F7901E0E82974856A7
h  = 1
```

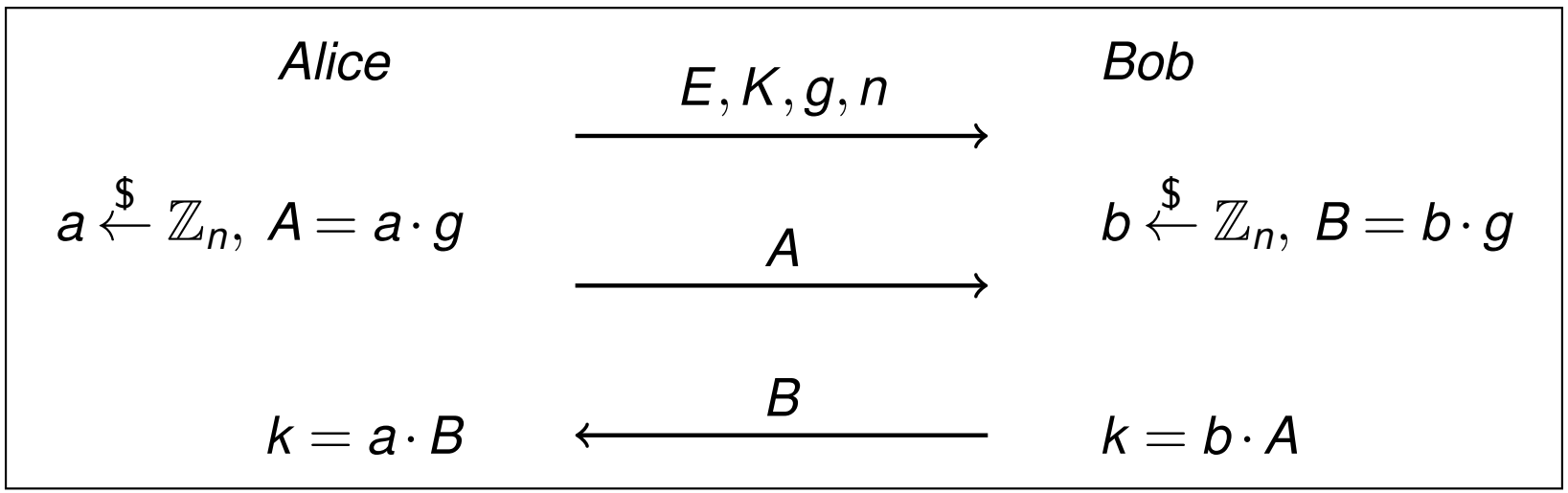
Elliptic Curve Diffie-Hellman (ECDH)

The multiplicative groups $\mathbb{Z}_p^*$ are often used for a Diffie-Hellman key exchange. However, over the last decade elliptic curves over $GF(p)$ have become increasingly popular, since they achieve a similar level of security with shorter keys (see below).

For a Diffie-Hellman key exchange with elliptic curves, the communication partners (say Alice and Bob) have to agree on a finite field $GF(p)$, an elliptic curve E over $GF(p)$, and a base point g generating a group $G \subset E(GF(p))$ of order n. Usually, they would choose a standard elliptic curve domain parameters.

An eavesdropper, who knows the public keys A or B as well as the elliptic curve and its domain parameters, should not be able to derive any information on the shared secret key k if the Diffie-Hellman problem is hard in G.

Elliptic Curve Diffie-Hellman Key Exchange



Elliptic-Curve Diffie-Hellman key exchange between Alice and Bob.

For uniform output, the hash $H(k)$ of the x-coordinate of the point k is used as shared secret key. Note that the y-coordinate of a point is (up to a sign) determined by the x-coordinate.

Example

Alice and Bob agree on the curve $y^2 = x^3 + 3x + 5$ over $GF(19)$, as well as the base point $g = 2 \cdot (1,3) = (18,18)$. The point g has order 13.

Alice chooses the secret key $a = 2$ and computes the public key

$$A = a \cdot g = 2 \cdot (18,18) = (11,18).$$

Bob chooses the secret key $b = 5$ and computes the public key

$$B = b \cdot g = 5 \cdot (18,18) = (0,9).$$

They exchange the public keys A and B. Alice obtains the shared secret key by computing

$$k = a \cdot B = 2 \cdot (0,9) = (9,18).$$

Bob computes the same key

$$k = b \cdot A = 5 \cdot (11,18) = (9,18).$$

ECDSA

The *Elliptic Curve Digital Signature Algorithm* (ECDSA) was standardized (by the American NIST) as part of the Digital Signature Standard (DSS) (see last chapter).

Choose elliptic curve domain parameters, i.e., a finite field $GF(p)$, an elliptic curve E over $GF(p)$, and a base point g generating a group $G \subset E(GF(p))$ of prime order n.

Definition of ECDSA

Definition

- Let $g \in E(GF(p))$ be the chosen base point of prime order n. Choose a uniform $x \in \mathbb{Z}_n$ and set $y = x \cdot g$. The private key is x and the public key is y. Furthermore, choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_n$.
- The randomized signature algorithm takes a message m and the private key x as input. Choose a uniform $k \in \mathbb{Z}_n$. Define $r = k \cdot g \mod n$, where only the x-coordinate is used. Then compute $s = k^{-1}(H(m) + xr) \mod n$. If $r = 0$ or $s = 0$ then start again with a new choice of k. Output the signature (r, s).
- The verification algorithm takes the public key y, a message m and a signature (r, s). The algorithm outputs 1 (valid) if

$$r = (H(m)s^{-1}) \cdot g + (rs^{-1}) \cdot y \mod n,$$

where only the x-coordinate is used. If both sides do not match or if the point at infinity is obtained, then output 0 (invalid).

Other Applications of Elliptic Curve Cryptography

Elliptic curves also have other applications in cryptography:

- Diffie-Hellman Key Encapsulation Mechanism (KEM).
- Elliptic Curve Integrated Encryption Scheme (ECIES).
- Pairing based cryptography, e.g. for identity-based encryption and BLS-signatures. A bilinear pairing is a map

$$e: \ E(GF(p)) \times E(GF(p)) \rightarrow GF(p^k)^*.$$

Surprisingly, elliptic curves can also be used to factorize large integers and to attack RSA if the modulus is relatively small, say 50-60 decimal digits.

- Elliptic Curve Factoring Method (ECM)

Security of Elliptic Curve Cryptography

The *security* of elliptic curve schemes relies on the hardness of the discrete-logarithm (DL) problem. A major difference to the multiplicative group is that the known sub-exponential algorithms for $\mathbb{Z}_p^*$ cannot be applied to elliptic curves.

If $n = ord(g)$ is a prime number, then the best known algorithms for computing discrete logarithms on elliptic curves are *Babystep-Giantstep* and *Pollard's* ρ*-method for logarithms*. Their complexity is $O(\sqrt{n})$, which is *exponential* in size (n).

However, the elliptic curve domain parameters must be carefully chosen in order to prevent certain types of attacks.

Furthermore, large *quantum computers* (which are not yet available) can break elliptic curve cryptography.

Comparison

The following table shows comparable security strengths for different algorithms and their key lengths (in bits). The table compares symmetric encryption schemes such as AES (key length), RSA (size of the modulus N), DH and DSA using a subgroup of order q in $\mathbb{Z}_p^*$ (size of p, size of q), as well as ECDH and ECDSA (size of n, where n is the order of the group generated by the base point).

Symmetric Encryption	RSA	DH, DSA	ECDH, ECDSA
80	1024	1024, 160	$160-223$
112	2048	2048, 224	$224-255$
128	3072	3072, 256	$256-383$
192	7680	7680, 384	$384-511$
256	15360	15360, 512	$512+$

Comparable security strengths of algorithms and key lengths.

Source: NIST SP 800-57 Part 1, Rev. 5, Table 2.