TABLE 4.2. Addition on an elliptic curve mod p.

Given: curve E: $y^2 = x^3 + ax + b \pmod p$
$P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ on E
Find: $P_3 = (x_3, y_3) = P_1 + P_2$
Algorithm:
$x_3 = m^2 - x_1 - x_2 \pmod p$
$y_3 = m(x_1 - x_3) - y_1 \pmod p$
where $m = \begin{cases} (y_2 - y_1) \cdot (x_2 - x_1)^{-1} \bmod p & \text{if } P_1 \neq P_2 \\ (3x_1^2 + a) \cdot (2y_1)^{-1} \bmod p & \text{if } P_1 = P_2 \end{cases}$
Special case 1: If $m = \infty$ then $P_3 = \infty$
Special case 2: $\infty + P = P$ for all P

4.5.2 ECC Diffie-Hellman

Now that we can do addition on elliptic curves, let's consider the ECC version of Diffie-Hellman. The public information consists of a curve and a point on the curve. We'll select the curve

$$y^2 = x^3 + 11x + b \pmod{167} \tag{4.7}$$

leaving b to be determined momentarily. Next, we can select any point (x, y) and determine b so that this point lies on the resulting curve. In this case, we'll choose, say $(x, y) = (2, 7)$. Then substituting $x = 2$ and $y = 7$ into *equation 4.7*, we find $b = 19$. Now the public information is

$$\text{Public: Curve } y^2 = x^3 + 11x + 19 \pmod{167} \text{ and point } (2, 7) \tag{4.8}$$

Alice and Bob each must select their own secret multipliers.[3] Suppose Alice selects $A = 15$ and Bob selects $B = 22$. Then Alice computes

$$A(2, 7) = 15(2, 7) = (102, 88)$$

where all arithmetic is done on the curve in *equation 4.8*. Alice sends this result to Bob. Bob computes

$$B(2, 7) = 22(2, 7) = (9, 43)$$

which he sends to Alice. Now Alice multiplies the value she received from Bob by her secret A, that is,

$$A(9, 43) = 15(9, 43) = (131, 140).$$

Similarly, Bob computes

$$B(102, 88) = 22(102, 88) = (131, 140)$$

and Alice and Bob have established a shared secret, suitable for use as a symmetric key. Note that this works since $AB(2, 7) = BA(2, 7)$. The security of this method rests on the fact that, although Trudy can see $A(2, 7)$ and $B(2, 7)$, she (apparently) must find A or B before she can determine the shared secret. As far as is known, this elliptic curve version of DH is as difficult to break as the regular DH. Actually, for a given number of bits, the elliptic curve version is harder to break, which allows for the use of smaller values for an equivalent level of security.

All is not lost for Trudy. She can take some comfort in the fact that the ECC version of DH is just as susceptible to a MiM attack as the usual Diffie-Hellman key exchange.

There are many good sources of information on elliptic curves. See [192] for a readable treatment and [28] for more of the mathematical details.

4.6 PUBLIC KEY NOTATION

Before discussing the uses of public key crypto, we need to consider the issue of notation. Since public key crypto uses two keys per user, adapting the notation that we used for symmetric key crypto would be awkward. In addition, a digital signature is an encryption (with the private key), but the same operation is a decryption when applied to ciphertext.

We'll adopt the notation used in [122] for public key encryption, decryption, and signing:

- Encrypt message M with Alice's public key: $C = \{M\}_{\text{Alice}}$
- Decrypt ciphertext C with Alice's private key: $M = [C]_{\text{Alice}}$
- Signing is the same operation as decrypting, so the notation for Alice signing message M is $S = [M]_{\text{Alice}}$, where S is the signed message

Since encryption and decryption are inverse operations,

$$[\{M\}_{\text{Alice}}]_{\text{Alice}} = \{[M]_{\text{Alice}}\}_{\text{Alice}} = M.$$

Never forget that the public key is public. As a result, anyone can compute $\{M\}_{\text{Alice}}$. On the other hand, the private key is private, so only Alice has access to her private key. As a result, only Alice can compute $[C]_{\text{Alice}}$ or $[M]_{\text{Alice}}$. The implication is that anyone can encrypt a message for Alice, but only Alice can decrypt the ciphertext. In terms of signing, only Alice can sign M, but, since the public key is public, anyone can verify

You may be wondering why we would use DH to establish a symmetric key if we already have a shared symmetric key (as in *1*) or a key pair (as in *2* and *3*). This is an excellent question to which we'll give an excellent answer when we discuss protocols in Chapters 9 and 10.

4.5 ELLIPTIC CURVE CRYPTOGRAPHY

"Elliptic curve" is not a particular cryptosystem. Instead, elliptic curves simply provide another way to perform the complex mathematical operations required in public key cryptography. For example, there is an elliptic curve version of Diffie-Hellman.

The advantage of elliptic curve cryptography (ECC) is that fewer bits are needed for the same level of security as in the non-elliptic curve case. On the down side, elliptic curves are more complex, and, as a result, mathematics on elliptic curves is somewhat more expensive. But overall, elliptic curves appear to offer a computational advantage. For this reason, ECC is particularly popular in resource-constrained environments such as handheld devices.

An elliptic curve E is the graph of a function of the form

$$E: \quad y^2 = x^3 + ax + b$$

together with a special point at infinity, denoted ∞. The graph of a typical elliptic curve appears in Figure 4.3.

4.5.1 Elliptic Curve Math

Figure 4.3 also illustrates the method used to find the sum of two points on an elliptic curve. To add the points P_1 and P_2, a line is drawn through the two points. This line

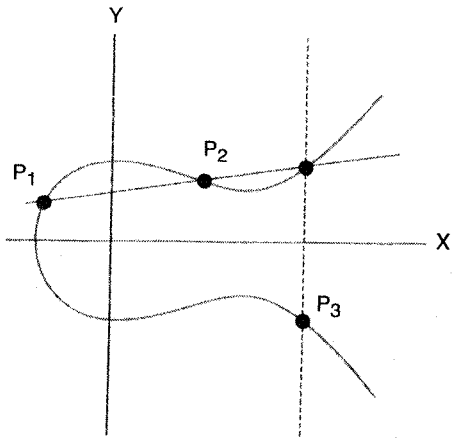

Figure 4.3. An elliptic curve.

TABLE 4.1. Points on the curve $y^2 = x^3 + 2x + 3 \pmod 5$.

$x = 0$	$\Longrightarrow$	$y^2 = 3 \Longrightarrow$ no solution mod 5
$x = 1$	$\Longrightarrow$	$y^2 = 6 = 1 \Longrightarrow y = 1, 4 \bmod 5$
$x = 2$	$\Longrightarrow$	$y^2 = 15 = 0 \Longrightarrow y = 0 \bmod 5$
$x = 3$	$\Longrightarrow$	$y^2 = 36 = 1 \Longrightarrow y = 1, 4 \bmod 5$
$x = 4$	$\Longrightarrow$	$y^2 = 75 = 0 \Longrightarrow y = 0 \bmod 5$

usually intersects the curve in one other point. If so, this other point is reflected ab the x-axis to obtain the sum,

$$P_3 = P_1 + P_2$$

as illustrated in Figure 4.3. Addition is the only mathematical operation on elliptic cur that we'll require.

For cryptography, we require a discrete set of points. This is easily accomplis by adding "mod p" to the generic elliptic curve equation, that is,

$$y^2 = x^3 + ax + b \pmod p.$$

For example, consider the elliptic curve

$$y^2 = x^3 + 2x + 3 \pmod 5. \qquad (4$$

We can list all of the points (x, y) on this curve by substituting for the possible val of x and solving for corresponding y value or values. Doing so, we obtain the result Table 4.1.

Then the points on the elliptic curve in *equation 4.5* are

$$(1, 1)\ (1, 4)\ (2, 0)\ (3, 1)\ (3, 4)\ (4, 0) \text{ and } \infty. \qquad (4$$

The algorithm for adding two points on an elliptic curve appears in Table 4.2. L apply the algorithm in Table 4.2 to find the points $P_3 = (1, 4) + (3, 1)$ on the curve *equation 4.5*. First, we compute

$$m = (1 - 4)/(3 - 1) = -3 \cdot 2^{-1} = -3 \cdot 3 = 1 \bmod 5.$$

Then

$$x_3 = 1^2 - 1 - 3 = -3 = 2 \bmod 5$$

and

$$y_3 = 1(1 - 2) - 4 = -5 = 0 \bmod 5.$$

Therefore, on the curve $y^2 = x^3 + 2x + 3 \pmod 5$, we have $(1, 4) + (3, 1) = (2,$ Note that $(2, 0)$ is also on the on the curve in *equation 4.5*, as indicated in *equation*