How to Encode a bitstring as a point on an Elliptic Curve

One notices that when you plug in a value for x in the equation for an elliptic curve, we find the following options for the solutions in y:

1. There are two values of y that satisfy the equation.
2. There are no values of y that satisfy the equation.
3. There is one value of y (0) that satisfies the equation.

Thus, our initial thought would be that we can't encode an arbitrary bitstring as the x-coordinate of a point on an elliptic curve because for some values of x (if we interpret a bitstring in binary as usual) no such point would occur. It is not feasible for us to not have some bitstrings we are unable to convey in plaintext.

But, what if it's the case that many x values (say roughly 50%) have two matching y values? (This is indeed the case.) Then one creative idea is as follows:

Say we want to transmit a 512-bit block. Imagine an elliptic curve where the prime number is 528 bits. What we could do is assign our least significant 512 bits for the x-value as the block we want to transmit:

----------$b_{16}b_{17}...b_{527}$

Then, we have 16 more bits to fill in. We can do either of the following:

1. Fill them in randomly, OR
2. Fill them in, in order (more on this later)

Say we try

0000000000000000$b_{16}b_{17}...b_{527}$

and we come to find out that there's no y value that satisfies the equation. Then we try

0000000000000001$b_{16}b_{17}...b_{527}$

followed by

0000000000000010$b_{16}b_{17}...b_{527}$

and so on.

If each of these has a 50% of success (which is actually the case), then the probability our first try succeeds is 50%, probability the first or second shot succeeds is 75%, the probability one of our first three shots succeeds is 87.5%, and so forth. More generally, the probability that k different choices of bits all fail in a row is $(\frac{1}{2})^k$, meaning the probability of success is $1 - (\frac{1}{2})^k$. Notice that

with 16 extra bits we have 2^{16} choices of strings, meaning our chance of failure (that none of the choices for these 16 bits will work is $(\frac{1}{2})^{65536}$, since 2^{16} = 65536. So, really for all practical purposes, we are unlikely to need more than 8 extra bits as the probability of failure in that case would be is $(\frac{1}{2})^{256}$, since 2^8 = 256.

Thus, this is precisely how we will end up encoding bitstrings via Elliptic Curves. We'll leave some buffer bits that aren't part of the message and fill these in with different choices until we find one that works. Assuming we leave in enough buffer bits, with extremely high probability (just like Miller-Rabin), we should be able to find a choice that works. The recipient will know to discard the bits that aren't part of the message.

How to Determine if $y^2 = x^3 + ax + b \pmod p$ has a solution for y for a given x

There is an advanced definition in Number Theory that addresses the following relevant question:

Does $y^2 \equiv c \ (mod\ p)$ have a solution for a given c and p, where p is prime?

By definition, the Legendre Symbol is the answer to this question. It's defined as follows:

$$\left(\frac{c}{p}\right) = 1, if\ \exists y | y^2 \equiv c \ (mod\ p), p\ is\ prime, c \neq 0$$
$$\left(\frac{c}{p}\right) = -1, if\ \nexists y | y^2 \equiv c \ (mod\ p), p\ is\ prime, c \neq 0$$
$$\left(\frac{c}{p}\right) = 0, if\ \ c = 0$$

So, now, our final question is: How do we compute the Legendre Symbol for a given prime p and integer c?

I am not including the proof here, but Euler figured out that $\left(\frac{c}{p}\right) = c^{\frac{(p-1)}{2}} \ (mod\ p)$. Notice from Fermat we know that $c^{p-1} \equiv 1 \ (mod\ p)$ and that $c^{\frac{(p-1)}{2}} \equiv \pm 1 \ (mod\ p)$.

Now, let's consider the case that $c^{\frac{(p-1)}{2}} \equiv 1 \ (mod\ p)$, which, according to Euler, indicates that there exists a solution for y in the equation $y^2 \equiv c \ (mod\ p)$.

Take that equation above and multiply both sides by c:

$$c^{\frac{(p-1)}{2}} \equiv 1 \ (mod\ p)$$

$$(c^1)c^{\frac{(p-1)}{2}} \equiv (c)1 \ (mod\ p)$$

$$c^{\frac{(p-1)}{2}+1} \equiv (c)1 \ (mod\ p)$$

$$c^{\frac{(p-1)}{2}+\frac{2}{2}} \equiv (c)\ (mod\ p)$$

$$c^{\frac{(p-1+2)}{2}} \equiv (c)\ (mod\ p)$$

$$c^{\frac{(p+1)}{2}} \equiv (c)\ (mod\ p)$$

Now, let $y = c^{\frac{(p+1)}{4}}$. This means that $y^2 \equiv \left(c^{\frac{(p+1)}{4}}\right)^2 \equiv c^{\frac{(p+1)}{2}} \equiv c\ (mod\ p)$, which means we've found our solution to y, so long as we're allowed to make the computation $y = c^{\frac{(p+1)}{4}}\ (mod\ p)$. This computation is allowed if and only if 4 | (p + 1), which would mean that so long as $p \equiv 3\ (mod\ 4)$, then we can find a corresponding solution y to the equation $y^2 \equiv c\ (mod\ p)$.

Now, we can clearly delineate our steps to take a bitstring s and encode it as a point on the elliptic curve $E_p(a, b)$, **where p is a prime equivalent to 3 mod 4:**

1. Set ExtraBits = 00...0

2. While not done do

    2a. Set x = ExtraBits + s, where + is concatenation
    2b. Compute $c = x^3 + ax + b \bmod p$, reducing the answer to be in between 0 and p – 1.
    2c. Compute $c^{\frac{(p-1)}{2}}\ (mod\ p)$.

    2d. If the result from 2c equals 1 do

        2d(i). The point on the curve to send is $(x, c^{\frac{(p+1)}{4}})$.
        2d(ii). Return this and/or break out of the loop.

    2e. Add 1 to ExtraBits.

Example Encoding (by hand and calculator)

Let's use the curve $E_{79}(3, 4)$. Let's just make sure this is a valid curve by making sure that $4a^3 + 27b^2$ isn't divisible by p:

$4(3)^3 + 27(4)^2 = 108 + 432 = 540 \equiv 66 \pmod{79}$, so we are good.

Let's attempt to send a 4-bit message, using two bits as our extra bits.

Let our plaintext message x = 13 (1101 in binary).

Here is a list of the x values we'll try to send, listed in binary:

001101
011101
101101
111101

The values of x we are trying, in sequence are: 13, 29, 45 and 61.

Checking x = 13

Step 2b: Compute $x^3 + ax + b \pmod{p}$

$13^3 + 3(13) + 4 \equiv 28 \pmod{79}$ (Computed on calculator, value w/o mod is 2240.)

Step 2c: Compute $c^{\frac{(p-1)}{2}} \ (mod\ p)$, so 28^{39} mod 79, using the calculator (or IDLE), we get that this is equivalent to 78 (mod 79), or -1, so this value of x fails.

Checking x = 29

Step 2b: Compute $x^3 + ax + b \pmod{p}$

$29^3 + 3(29) + 4 \equiv 69 \pmod{79}$ (Computed on calculator)

Step 2c: Compute $c^{\frac{(p-1)}{2}} \ (mod\ p)$, so 69^{39} mod 79, using the calculator (or IDLE), we get that this is equivalent to 78 (mod 79), or -1, so this value of x fails.

Checking x = 45

Step 2b: Compute $x^3 + ax + b \pmod{p}$

$45^3 + 3(45) + 4 \equiv 19 \pmod{79}$ (Computed on calculator)

Step 2c: Compute $c^{\frac{(p-1)}{2}} \ (mod\ p)$, so 19^{39} mod 79, using the calculator (or IDLE), we get that this is equivalent to 1 (mod 79), so now we found a value that works!

Step 2d: Compute $y = c^{\frac{(p+1)}{4}} \pmod{p}$, so 19^{20} mod p is equivalent to 16 (mod 79)

The point to send is (45, 16).

Let's verify this is on the curve!

$y^2 \equiv 16^2 \equiv 256 \equiv 19 \ (mod\ 79)$

$x^3 + 3x + 4 \equiv 45^3 + 3(45) + 4 \equiv 91125 + 135 + 4 \equiv 91264 \equiv 19 \ (mod\ 79)$

Indeed, the point is on the curve.

Thus, in our El-Gamal scheme, Bob would encode the point (45, 16) and Alice would recover this point. Alice would know that only the 4 least significant bits of the x-coordinate were part of the message and she would compute 45%16 to arrive at the correct plaintext of 13.

A more succinct way of viewing this information is in a chart format:

x	$c = x^3 + 3x + 4 \bmod 79$	$c^{\frac{79-1}{2}} mod\ 79$	$c^{\frac{79+1}{4}} mod\ 79$
13	28	78	
29	69	78	
45	19	1	16