For ElGamal, the message space actually is $\{1,2,\ldots,p-1\}$. $m=0$ is not permitted.

That's, of course, no practical issue. For instance, we could simply identify $\{1,2,\ldots,p-1\}$ with $\{0,1,\ldots,p-2\}$ by adding/subtracting 1.

Example 161. Bob's public ElGamal key is $(p,g,h)=(23,10,11)$.

(a) Encrypt the message $m=5$ ("randomly" choose $y=2$) and send it to Bob.

(b) Encrypt the message $m=5$ ("randomly" choose $y=4$) and send it to Bob.

(c) Break the cryptosystem and determine Bob's secret key.

(d) Use the secret key to decrypt $c=(8,7)$.

(e) Likewise, decrypt $c=(18,19)$.

Solution.

(a) The ciphertext is $c=(c_1,c_2)$ with $c_1=g^y \pmod{p}$ and $c_2=h^y m \pmod{p}$.
Here, $c_1=10^2\equiv 8 \pmod{23}$ and $c_2=11^2\cdot 5\equiv 6\cdot 5\equiv 7 \pmod{23}$. Hence, the ciphertext is $c=(8,7)$.

(b) Now, $c_1=10^4\equiv 18 \pmod{23}$ and $c_2=11^4\cdot 5\equiv 13\cdot 5\equiv 19 \pmod{23}$ so that $c=(18,19)$.

(c) We need to solve $10^x\equiv 11 \pmod{23}$. This yields $x=3$.
(Since we haven't learned a better method, we just try $x=1,2,3,\ldots$ until we find the right one.)

(d) We decrypt $m=c_2 c_1^{-x} \pmod{p}$.
Here, $m=7\cdot 8^{-3}\equiv 7\cdot 4\equiv 5 \pmod{23}$.
[$8^{-1}\equiv 3 \pmod{23}$, so that $8^{-3}\equiv 3^3\equiv 4 \pmod{23}$. Or, use Fermat: $8^{-3}\equiv 8^{19}\equiv 4 \pmod{23}$.]

(e) In this case, $m=19\cdot 18^{-3}\equiv 19\cdot 16\equiv 5 \pmod{23}$.

Example 162. (homework) If Bob selects $p=23$, how many possible choices does he have for g? Which are these?

Solution. g must be a primitive root modulo p.

- Recall that, modulo a prime p, there always exists a primitive root g.
Here, the smallest primitive root is $g=5$. (Or, we could just use $g=10$ from the previous example.)
To check that, we need to verify that the order of $5 \pmod{23}$ is 22. Since the order must divide 22, it is enough to check that $5^2\not\equiv 1 \pmod{23}$ and $5^{11}\not\equiv 1 \pmod{23}$.

- By definition, g has order $p-1$. Then, all other invertible residues can be expressed as g^a, which has order $(p-1)/\gcd(p-1,a)$. In order for g^a to be a primitive root, we therefore need $\gcd(p-1,a)=1$. There are $\phi(p-1)=\phi(22)=10$ such values a in the range $1,2,\ldots,22$.

- The possible 10 values for a are $1,3,5,7,9,13,15,17,19,21$.
The corresponding 10 primitive roots are $5^1,5^3,5^5,5^7,\ldots \pmod{23}$. Explicitly computing these powers, the primitive roots are $5,7,10,11,14,15,17,19,20,21 \pmod{23}$.

Armin Straub
straub@southalabama.edu

We indicated that the security of ElGamal depends on the difficulty of computing discrete logarithms. Here is a more precise statement.

Theorem 163. Decrypting c to m in ElGamal is exactly as difficult as the **computational Diffie–Hellman problem** (CDH).

The CDH problem is the following: given $g, g^x, g^y \pmod{p}$, find $g^{xy} \pmod{p}$. It is believed to be hard.

Proof. Recall that the public key is $(p, g, h) = (p, g, g^x)$. The ciphertext is $c = (g^y, h^y m) = (g^y, g^{xy} m)$.
Hence, determining m is equivalent to finding g^{xy}.
Since $g, g^x, g^y \pmod{p}$ are known, this is precisely the CDH problem. □

Example 164. In fact, even the **decisional Diffie–Hellman problem** (DDH) is believed to be difficult.

The DDH problem is the following: given $g, g^x, g^y, r \pmod{p}$, decide whether $r \equiv g^{xy} \pmod{p}$. Obviously, this is simpler than the CDH problem, where g^{xy} needs to be computed. Yet, it, too, is believed to be hard.

Comment. Well, at least it is hard (modulo p) if we always want to do better than guessing.

Here's how we can sometimes do better than guessing: if g^x or g^y are quadratic residues (this is actually easy to check modulo primes p using quadratic reciprocity and the Legendre symbol), then g^{xy} is quadratic residue (why?!). Hence, if r is not a quadratic residue, we can conclude that $r \not\equiv g^{xy}$.

7.1 Diffie–Hellman key exchange

The key idea that makes ElGamal encryption work is that Alice (her private secret is y) and Bob (his private secret is x) actually share a secret: g^{xy}

Since g^x is publicly known, Alice can compute $g^{xy} = (g^x)^y$ using her secret y.
Similarly, since g^y is known from the ciphertext, Bob can compute $g^{xy} = (g^y)^x$ using his secret x.

(Diffie–Hellman key exchange)

- Alice or Bob choose a prime p and a primitive root $g \pmod{p}$.
- Bob randomly selects a secret integer x and reveals $g^x \pmod{p}$ to everyone.
  Alice randomly selects a secret integer y and reveals $g^y \pmod{p}$ to everyone.
- As above, Alice and Bob now share the secret $g^{xy} \pmod{p}$.

Why is this secure? We need to see why eavesdropping Eve cannot (simply) obtain the secret $g^{xy} \pmod{p}$.
She knows $g, g^x, g^y \pmod{p}$ and needs to find $g^{xy} \pmod{p}$.
This is precisely the CDH problem, which is believed to be hard.

Example 165. (homework) You are Eve. Alice and Bob select $p = 53$ and $g = 5$ for a Diffie–Hellman key exchange. Alice sends 43 to Bob, and Bob sends 20 to Alice. What is their shared secret?

Solution. Let's crack Alice's secret y (you can also attack Bob).
For that, we need to find y such that $5^y = 43 \pmod{53}$.
We try all possibilities: $5^2 = 25$, $5^3 \equiv 19$, $5^4 \equiv 19 \cdot 5 \equiv -11$, $5^5 \equiv -11 \cdot 5 \equiv -2$, $5^6 \equiv -2 \cdot 5 \equiv -10 \equiv 43 \pmod{53}$.
Hence, Alice's secret is $y = 6$. The shared secret is $20^6 \equiv 9 \pmod{53}$.