The ElGamal public key cryptosystem and discrete logarithms

Whereas the security of RSA relies on the difficulty of factoring, the security of ElGamal and Diffie–Hellman relies on the difficulty of computing discrete logarithms.

Discrete logarithms

Suppose $b = a^x \pmod{N}$. Finding x is called the **discrete logarithm problem** mod N. If N is a large prime p, then this problem is believed to be difficult.

Note. If $b = a^x$, then $x = \log_a(b)$. Here, we are doing the same thing, but modulo N. That's why the problem is called the discrete logarithm problem.

Example 170. Find x such that $4 \equiv 3^x \pmod{7}$.

Solution. We have seen in Example 155 that 3 is a primitive root modulo 7. Hence, there must be such an x. Going through the possibilities ($3^2 \equiv 2$, $3^3 \equiv 6$, $3^4 \equiv 4$), we find $x = 4$, because $3^4 \equiv 4 \pmod{7}$.

Example 171. Find x such that $3 \equiv 2^x \pmod{101}$.

Solution. Let us check that the solution is $x = 69$. Indeed, a quick binary exponentiation confirms that $2^{69} \equiv 3 \pmod{101}$. (Do it!)

The point is that it is actually (believed to be) very difficult to compute these **discrete logarithms**. On the other hand, just like with factorization, it is super easy to verify the answer if somebody tells us the answer.

Comment. We can check that 2 is a primitive root modulo 101. That is, $2 \pmod{101}$ has (multiplicative) order 100. That means every equation $2^x \equiv a \pmod{101}$, where $a \not\equiv 0$, has a solution.

Diffie–Hellman key exchange

(Diffie–Hellman key exchange)

- Alice and Bob select a large 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.
- Alice and Bob now share the secret $g^{xy} \pmod{p}$.
  Indeed, Alice can compute $g^{xy} = (g^x)^y$ using the public g^x and her secret y.
  Likewise, Bob can compute $g^{xy} = (g^y)^x$ using the public g^y and his secret x.

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 the **computational Diffie–Hellman problem** (CDH), which is believed to be hard (it would be easy if we could compute discrete logarithms).

Example 172. 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. If Alice's secret is y and Bob's secret is x, then $5^y \equiv 43$ and $5^x \equiv 20 \pmod{53}$.

Since we haven't learned a better method, we just compute $5^2, 5^3, \ldots$ until we find 43 or 20:

$5^2 = 25$, $5^3 \equiv 19$, $5^4 \equiv 19 \cdot 5 \equiv -11$, $5^5 \equiv -11 \cdot 5 \equiv -2$, $5^6 \equiv -2 \cdot 5 \equiv -10 \equiv 43 \pmod{53}$.

Hence, Alice's secret is $y = 6$. The shared secret is $20^6 \equiv 9 \pmod{53}$.

Note. We don't need to find Bob's secret. [It is $x = 11$.]

Proposed by Taher ElGamal in 1985

The original paper is actually very readable: https://dx.doi.org/10.1109/TIT.1985.1057074

(ElGamal encryption)

- Bob chooses a prime p and a primitive root $g \pmod{p}$.
  Bob also randomly selects a secret integer x and computes $h = g^x \pmod{p}$.
- Bob makes (p, g, h) public. His (secret) private key is x.
- To encrypt, Alice first randomly selects an integer y.
  Then, $c = (c_1, c_2)$ with $c_1 = g^y \pmod{p}$ and $c_2 = h^y m \pmod{p}$.
- Bob decrypts $m = c_2 c_1^{-x} \pmod{p}$.

Why does decryption work? $c_2 c_1^{-x} = (h^y m)(g^y)^{-x} = ((g^x)^y m)(g^y)^{-x} = m \pmod{p}$

More conceptually, the key idea (featured in Diffie–Hellman) 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}

Note that encryption is just multiplying m with the shared secret $h^y = g^{xy}$. Likewise, decryption is division by the shared secret $c_1^x = g^{xy}$.

Comment. 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.

Comment. p and g don't have to be chosen randomly. They can be reused. In fact, it is common to choose p to be a "safe prime" (see next comment), with specific pre-selected choices listed, for instance, in RFC 3526.

Advanced comment. Note that in order to check whether g is a primitive root modulo p, we need to be able to factor $p-1$, which in general is hard (2 is an obvious factor, but other factors are typically large and, in fact, we need them to be large in order for the discrete logarithm problem to be difficult). It is therefore common to start with a prime n and then see if $2n+1$ is prime as well, in which case we select $p = 2n+1$. Such primes p [primes such that $(p-1)/2$ is prime, too] are called **safe primes** (more later).

On the other hand, g doesn't necessarily have to be a primitive root. However, we need the group generated by g (the elements 1, g, g^2, g^3, $\ldots$) to be large. For more fancy cryptosystems, we can even replace these groups with other groups such as those generated by elliptic curves.

Example 173. Bob chooses the prime $p = 31$, $g = 11$, and $x = 5$. What is his public key?

Solution. Since $h = g^x \pmod{p}$ is $h \equiv 11^5 \equiv 6 \pmod{31}$, the public key is $(p, g, h) = (31, 11, 6)$.

Comment. Bob's secret key is $x = 5$. In principle, an attacker can compute x from $11^x \equiv 6 \pmod{31}$. However, this requires computing a discrete logarithm, which is believed to be difficult if p is large.

Example 174. Bob's public ElGamal key is $(p, g, h) = (31, 11, 6)$.

(a) Encrypt the message $m = 3$ ("randomly" choose $y = 4$) and send it to Bob.

(b) Determine Bob's private key from his public key.

(c) Using Bob's private key, decrypt $c = (9, 13)$.

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, $c_1=11^4\equiv 9 \pmod{31}$ and $c_2=6^4\cdot 3\equiv 13 \pmod{31}$. Hence, the ciphertext is $c=(9,13)$.

(b) To find Bob's secret key x, we need to solve $11^x\equiv 6 \pmod{31}$. This yields $x=5$.
(Since we haven't learned a better method, we just try $x=1,2,3,\ldots$ until we find the right one.)
Comment. Alternatively, after having done the first part, we know that $m=c_2c_1^{-x} \pmod{p}$ takes the form $3=13\cdot 9^{-x} \pmod{31}$, which is equivalent to $9^x=13\cdot 3^{-1}\equiv 25 \pmod{31}$. While this also reveals $x=5$, there is an issue with this approach. Can you see it?
[The issue is that 9 (which is c_1 and could be anything) does not have to be a primitive root. In fact, 9 is not a primitive root modulo 31. Accordingly, $9^x\equiv 25 \pmod{31}$ does not have a unique solution: $x=20$ is another one (and does not correspond to Bob's private key).]

(c) We decrypt $m=c_2c_1^{-x} \pmod{p}$.
Here, $m=13\cdot 9^{-5}\equiv 3 \pmod{31}$.
Comment. One option is to compute $9^{-1}\equiv 7 \pmod{31}$, followed by $9^{-5}\equiv 7^5\equiv 5 \pmod{31}$ and, finally, $13\cdot 9^{-5}\equiv 13\cdot 5\equiv 3 \pmod{31}$. Another option is to begin with $9^{-5}\equiv 9^{25} \pmod{31}$ (by Fermat's little theorem).

Example 175. (extra) 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) To find Bob's secret key x, 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 $m=c_2c_1^{-x} \pmod{p}$.
Here, $m=7\cdot 8^{-3}\equiv 7\cdot 4\equiv 5 \pmod{23}$, as we knew from the first part.
[$8^{-1}\equiv 3 \pmod{23}$, 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, $m=19\cdot 18^{-3}\equiv 19\cdot 16\equiv 5 \pmod{23}$, as we knew from the second part.

Example 176. If Bob selects $p=23$ for ElGamal, how many possible choices does he have for g? Which are these?

Solution. g needs to be a primitive root modulo 23. Recall that, modulo a prime p, there are $\phi(\phi(p))=\phi(p-1)$ many primitive roots. Hence, Bob has $\phi(p-1)=\phi(22)=10$ choices for g.