7 The ElGamal public key cryptosystem and discrete logarithms

- Proposed by Taher ElGamal in 1985

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

- Whereas the security of RSA relies on the difficulty of factoring, the security of ElGamal relies on the difficulty of computing 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 156. Find x such that $4 \equiv 3^x \pmod 7$.

Solution. We have seen in Example 99 that 3 is a primitive root modulo 7. Hence, there must be such an x. Going through the possibilities, we find $x = 4$, because $3^4 \equiv 4 \pmod 7$.

Example 157. 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.

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

- Like RSA, ElGamal is terribly slow compared with symmetric ciphers like AES.

  Encryption under ElGamal requires two exponentiations (slower than RSA); however, these exponentiations are independent of the message and can be computed ahead of time if need be (in that case, encryption is just a multiplication, which is much faster than RSA). Decryption only requires one exponentiation (like RSA).

- In contrast to RSA, ElGamal is randomized. That is, a single plaintext m can be encrypted to many different ciphertexts.

  A drawback is that the ciphertext is twice as large as the plaintext.

  On the positive side, an attacker who might be able to guess potential plaintexts cannot (as in the case of vanilla RSA) encrypt these herself and compare with the intercepted ciphertext.

Example 158. (homework) 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 difficult if p is large.

Example 159. 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) **(homework)** Recall that Bob's secret private key is $x=5$. Use it to 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) 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 160. (homework) Bob's public ElGamal key is $(p,g,h)=(41,7,20)$.

(a) Encrypt the message $m=10$ ("randomly" choose $y=15$) and send it to Bob.

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

(c) Use the secret key to decrypt $c=(14,8)$.

Solution. (final answers only)

(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, $c_1=7^{15}\equiv 14 \pmod{41}$ and $c_2=20^{15}\cdot 10\equiv 8 \pmod{41}$. Hence, the ciphertext is $c=(14,8)$.

(b) We need to solve $7^x\equiv 20 \pmod{41}$. This yields $x=6$.
(Since we haven't learned a better method, you can just try $x=1,2,3,\ldots$ until you find the right one.)

(c) We decrypt $m=c_2c_1^{-x} \pmod p$.
Here, $m=8\cdot 14^{-6}\equiv 10 \pmod{41}$.