# Worksheet 23

1. Here is a fairly straightforward example of the Diffie-Hellman procedure.

(a) Let $p = 2609$ and $g = 7$ be a generator mod $p$. Use the Diffie-Hellman procedure to compute the shared secret between Alice and Bob using the private keys $a = 27$ and $b = 31$.

(b) Suppose that you're Eve the spy trying to listen in on Alice and Bob's conversation. You manage to intercept the messages 2394 and 715 exchanged between them. Use WolframAlpha to figure out what their shared secret key is.

2. ASCII is the standard way of converting symbols into numbers. Let $A = 65, B = 66, \ldots, Z = 90$, and let a space correspond to 32. (This is still not a particularly *secure* method for converting text into numbers, but this is a simple illustration of how your computer might do so without any other algorithm.)

   (a) Let $(m, e) = (4951760154835678088235319297, 18505676233006159663039548 77)$ be the public key. Convert the message "HELLO WORLD" into a number and using WolframAlpha, use the RSA method to encrypt your message.

   (b) Suppose that you are Eve the spy. Using WolframAlpha, compute the factorization of $m$ and then compute $\varphi(m)$. Then, compute the decryption key $d$. Use this to decode the intercepted message $(55799119760817384352725395, 31323399839857355784726744 02)$ that Bob sent to Alice, and convert it back to plain text (using a decimal to ASCII converter, perhaps).

   As you can see, the numbers used in this example are not even remotely difficult for a computer to break RSA with. The actual recommendation for primes used for serious applications of RSA are between 309 to 617 digits long!

3. RSA is a cryptosystem based off the idea that factoring is hard. Here is another cryptosystem called *ElGamal*, based off the hardness of the discrete logarithm problem. Suppose that Alice and Bob want to communicate. Alice first chooses an integer $d$ with $1 \leq d \leq p - 2$, which is her private key. Her public key is $(p, g, a)$ where $p$ is a prime, $g$ is a generator mod $p$, and $a \equiv g^d \bmod p$. To communicate with Alice, suppose Bob wants to send a short message $M$ (short meaning $M < p$). Bob sends it to Alice by choosing an integer $j$ (kept secret) and computing $C_1 \equiv g^j \bmod p$ and $C_2 \equiv a^j M \bmod p$. The message he sends Alice is $(C_1, C_2)$, which Alice can then decode by computing $C_2 C_1^{-d} \bmod p$.

   (a) Explain why this procedure actually works. That is, prove that $C_2 C_1^{-d} \equiv M \bmod p$.

   (b) Suppose that Alice has public key $(2147483647, 7, 1004798284)$, and Bob wishes to send the message "TEST" to Alice. Using ASCII, convert this into a number $M$ and use WolframAlpha to compute the message $(C_1, C_2)$ he sends Alice using $j = 15$.

   (c) Suppose you are Alice, so that you know your private key is $d = 101$. Using WolframAlpha, decipher the message that Bob sent you from part (a).

   Since ElGamal is based on the discrete logarithm problem, one benefit is that it can be converted into the setting of elliptic curves in order to make it even more secure!