Vanessa Heynes

Diffie Hellman

$g = 7, p = 61$
$g^x \bmod p = 30$
$g^y \bmod p = 17$

$7^x \bmod 61 = 30$
$7^y \bmod 61 = 17$

Values for x and y can be found through a python loop that checks all values from 1 to 61 (p) until the above conditions are met.

$x = 41, y = 23$

The shared secret is as follows: $7^{23x41} \bmod 61 = \mathbf{6}$

The process would have failed in the last step because finding values for x and y would get slower if Alice and Bob's chosen numbers, p and g, were much larger since those values act as upper bounds. Should there be multiple values of x and y that would satisfy the condition, then it would take even more time to find the secret number.

RSA

How did you figure out the message and how is it encoded?

Given that $n = 170171$ and $e = 17$, I can find values for p and q through brute force since $pq = n$. These values also need to be prime numbers. Without having to try all possible values, trying to find a value that satisfies $n \bmod p = 0$ can be done by starting from the floor of the square root of n which is 412. After finding p, dividing n by p will give q.

$p = 449, q = 379$

Next, I found d by using brute force, the values I already know, and the following property: $(e * d) \bmod ((p-1)(q-1)) = 1$

$d = 119537$

From here, I computed $M^d \bmod n$ for each of the blocks in the encrypted message. The resulting numbers/decrypted blocks when converted to binary reveal two ASCII characters which can then be read to spell out the secret message Alice sent to Bob.

The message was encoded by converting all the characters to binary and grouping them in adjacent pairs. Then, these paired values are converted to their integer values which are then encoded with Alice's public key using the following formula: $C(M) = M^e \bmod n$

What is the secret message?

Hi Bob. I'm walking from now on. Your pal, Alice.
https://foundation.mozilla.org/en/privacynotincluded/articles/its-official-cars-are-the-worst-product-category-we-have-ever-reviewed-for-privacy/

Show precisely where in your process you would have failed if the integers involved were much larger.

Since the values for d, p, q have to be smaller than n, should n be a much larger number, checking every integer below that boundary would become very inefficient. Numbers that require a high amount of bytes to be stored can also be hard for Python to handle if there isn’t enough memory to account for large enough numbers.

Explain, briefly, why the message encoding Alice used would be insecure even if Bob's keys involved larger integers.

With larger keys, decryption becomes a slower process which makes the communication between Alice and Bob weaker. Since Alice breaks up the message into smaller blocks before encrypting them, these characters can be matched to English without the encryption function.