Being Eve

Beshir Said

September 29, 2025

1 Diffie–Hellman

As Eve, we know that Alice and Bob agreed on $g = 5$ and $p = 103$, and that Alice sent Bob the number 10 while Bob sent Alice the number 71. Given these values, we can recover Alice's and Bob's secrets a and b by solving the discrete logarithms $5^a \equiv 10 \pmod{103}$ and $5^b \equiv 71 \pmod{103}$. Because the numbers are tiny, we can brute-force these to find $a = 45$ and $b = 67$; with larger integers, this step would be infeasible. Once we have the secrets, the shared key is $K = B^a \bmod p = A^b \bmod p$. Computing both ways gives $K = 71^{45} \bmod 103 = 31$ and $K = 10^{67} \bmod 103 = 31$, so Alice and Bob's shared key is 31.

2 RSA

As Eve, given Bob's public key $(e, n) = (17,\ 266{,}473)$ and the cipher text list, I first factored n as $266{,}473 = 439 \times 607$, computed $\lambda(n) = (439 - 1)(607 - 1) = 265{,}428$, and got $d = 187{,}361$. Using these values, I took the encrypted data sent from Alice and decrypted each cipher text integer with my d and n using this Python script pt = ([pow(c,d,n) for c in ct] and got ASCII values. Each decyrpted number is two characted packed together, so you have to split each into two byte values. I used the script to get the message: pt=[18533]; print([(m//256, m mod 256) for m in pt]).

The recovered message is: `Hey Bob, here's some cryptography history for you (https://en.wikipedia.org/wiki/The_Magic_Words_are_Squeamish_Ossifrage). Happy factoring, Alice.`

This only works here because n was tiny. On bigger RSA sizes the first step (factoring n) is infeasible, so you never get d; and even with big keys. Even with larger keys, the two-byte encoding is insuecure becuase identical 2 byte chunks encrypt identical cipher texts which allows for pattern recognition and this is a big security issue that could enable more attacks.