**[p, q, N, phi, e, publicKey, message, encryptedMsg, d, privateKey, decryptedMsg]**

**['23', '31', '713', '660', '61', '(713, 61)', '100', '565', '541', '(541, 713)', '100']**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p | q | N | ø | e | pub key | m | enc | d | pri key | dec |
| 23 | 31 | 713 | 660 | 61 | (713, 61) | 100 | 565 | 541 | (541, 713) | 100 |

**RSA Key Creation**

* Bob chooses two secret primes p = 23 and q = 31. Bob computes his public modulus
* Bob chooses a public encryption exponent e = 61 with the property that

**RSA Encryption**

* Alice converts her plaintext into an integer

satisfying .

* Alice uses Bob’s public key (N, e) = (713, 61) to compute
* Alice sends the ciphertext c = 565 to Bob.

**RSA Decryption**

* Bob knows , so he can solve

,

for *d* and find that .

* Bob takes the ciphertext and computes

,

.

The value that he computes is Alice’s message m = 100.

**[p, g, a, b, A, B, A', B']**

**['2333', '233', '97', '345', '1613', '357', '93', '93']**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **p** | **g** | **a** | **b** | **A** | **B** | **A’** | **B’** |
| **31** | **11** | **17** | **19** | **3** | **22** | **13** | **13** |