ElGamal and RSA Pseudocode

Mia Kelley-Lanser and Tenzin Sommer

1 Encryption

Algorithm 1 RSA Encryption

INPUT: A plaintext message in numerical form m, the public encryption key of the receiver e, the receiver's generated public modulus N

OUTPUT: The cipher text message in numerical form

1: **procedure** ENCRYPT(m, e, N):
2: **return** $m^e \ \% \ N$

Algorithm 2 ElGamal Encryption

INPUT: A plaintext message in numerical form m, a random numerical key k, the public key of the receiver $pubKey$, the chosen public prime modulus p, the chosen public generator g

OUTPUT: A pair of cipher text values (c_1, c_2)

1: **procedure** ENCRYPT($m, k, pubKey, p, g$):
2: $c_1 \leftarrow g^k \ \% \ p$
3: $c_2 \leftarrow m \cdot pubKey^k \ \% \ p$
4: **return** (c_1, c_2)

2 Decryption

Algorithm 3 RSA Decryption

INPUT: The cipher text numerical value c, the public encryption key of the receiver e, the receiver's generated public modulus N, the receiver's private primes p and q

OUTPUT: The decrypted message in numerical form

1: **procedure** DECRYPT(c, e, N, p, q):
2: $\quad d \leftarrow c^{-1} \ \% \ (p-1)(q-1)$
3: $\quad$ **return** $c^d \ \% \ N$

Algorithm 4 ElGamal Decryption

INPUT: A pair of cipher text values (c_1, c_2), the private key $privKey$, the chosen public prime modulus p, the chosen public generator g

OUTPUT: The decrypted message in numerical form

1: **procedure** DECRYPT$((c_1, c_2), privKey, p, g)$:
2: $\quad x \leftarrow (c_1^{privKey})^{-1} \ \% \ p$
3: $\quad$ **return** $(x \cdot c_2) \ \% \ p$