

CRYPTOSYSTEM 1. **RSA!**

Key Generation keygen.

Input: Security parameter  $n$ .

Output: secret key  $\mathbf{sk}$  and public key  $\mathbf{pk}$ .

1. Choose two distinct primes  $p$  and  $q$  at random with  $2^{(n-1)/2} < p, q < 2^{n/2}$ .
2.  $N \leftarrow p \cdot q$ ,  $L \leftarrow (p-1)(q-1)$ . [ $N$  is an  $n$ -bit number, and  $L = \varphi(N)$  is the value of Euler's totientfunction.]
3. Choose  $e \xleftarrow{\$} \{2, \dots, L-2\}$  at random, coprime to  $L$ .
4. Calculate the inverse  $d$  of  $e$  in  $\mathbb{Z}_L$ .
5. Publish the public key  $\mathbf{pk} = (N, e)$  and keep  $\mathbf{sk} = (N, d)$  as the secret key.

Encryption enc.

Input:  $x \in \mathbb{Z}_N$ ,  $\mathbf{pk} = (N, e)$ .

Output:  $\text{enc}_{\mathbf{pk}}(x) \in \mathbb{Z}_N$ .

6.  $y \leftarrow x^e$  in  $\mathbb{Z}_N$ .
7. Return  $\text{enc}_{\mathbf{pk}}(x) = y$ .

Decryption dec.

Input:  $y \in \mathbb{Z}_N$ ,  $\mathbf{sk} = (N, d)$ .

Output:  $\text{dec}_{\mathbf{sk}}(y) \in \mathbb{Z}_N$ .

8.  $x^* \leftarrow y^d$  in  $\mathbb{Z}_N$ .
9. Return  $\text{dec}_{\mathbf{sk}}(y) = x^*$ .

security parameter  $n$ ,  
distinct random primes  $p$  and  $q$  of  $n/2$  bits,  
 $N = pq$  of  $n$  bits,  
 $L = \varphi(N) = (p - 1)(q - 1)$ ,  
 $e, d \in \mathbb{Z}_L \setminus \pm 1$  with  $ed = 1$  in  $\mathbb{Z}_L$ ,  
plaintext  $x$ , ciphertext  $y$ , decryption  $x^*$ , all in  $\mathbb{Z}_N$ ,  
 $y = x^e, x^* = y^d$ .

Figure 1: The **RSA!** notation.

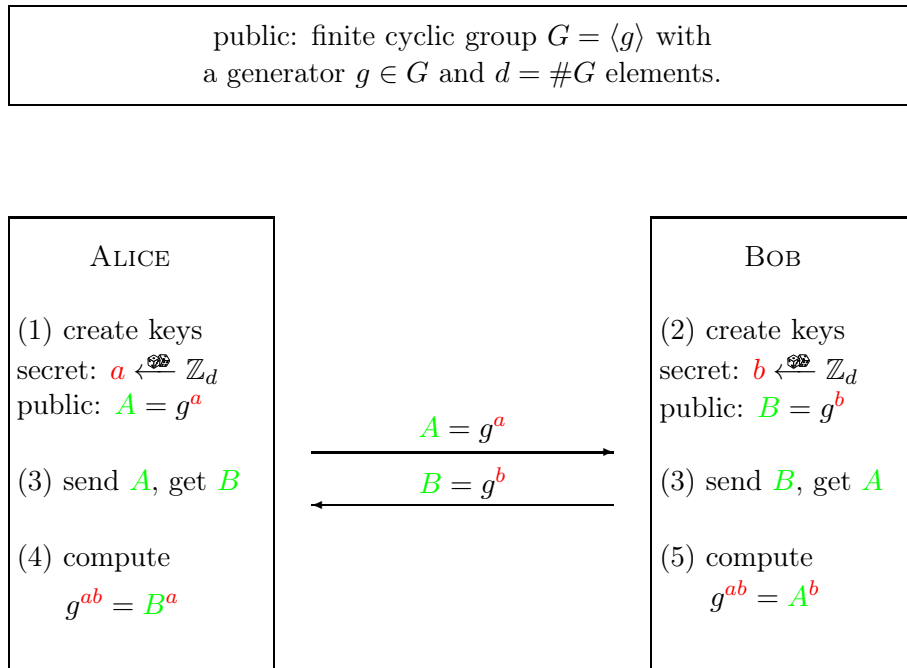


Figure 2: Diffie-Hellman key exchange.

PROTOCOL 2. ?? key exchange.

Key generation.

Input: security parameter  $n$ .

Output:  $G$ ,  $g$  and  $d$  as below.

1. Determine a description of a finite cyclic group  $G = \langle g \rangle$  with  $d = \#G$  elements and a generating element  $g$ , where  $d$  is an  $n$ -bit integer.

Key exchange.

2. ALICE chooses her secret key  $a \xleftarrow{\$} \mathbb{Z}_d$ . She computes her public key  $A \leftarrow g^a \in G$ .
3. BOB chooses his secret key  $b \xleftarrow{\$} \mathbb{Z}_d$ . He computes his public key  $B \leftarrow g^b \in G$ .
4. ALICE and BOB exchange their public keys  $A$  and  $B$ .
5. ALICE computes the common secret key  $k_A = B^a$ .
6. BOB computes the common secret key  $k_B = A^b$ .

EXAMPLE 3. Let  $G = \mathbb{Z}_{2579}^\times$  and  $g = 2 \in G$ .

Protocol 2 step 2. ALICE chooses her secret key  $a = 765$  and computes her public key  $A = 2^{765} = 949$  in  $G$ .

Protocol 2 step 3. BOB chooses his secret key  $b = 853$  and computes his public key  $B = 2^{853} = 435$  in  $G$ .

Protocol 2 step 4. ALICE and BOB exchange their public keys  $A = 949$  and  $B = 435$ .

Protocol 2 step 5. ALICE computes the common secret key  $k_A = B^{765} = 2424$  in  $G$ .

Protocol 2 step 6. BOB computes the common secret key  $k_B = A^{853} = 2424$  in  $G$ .

And lo and behold, the system works not only in general, but also in this particular case: ALICE and BOB share the key  $k_A = k_B$ .  $\diamond$

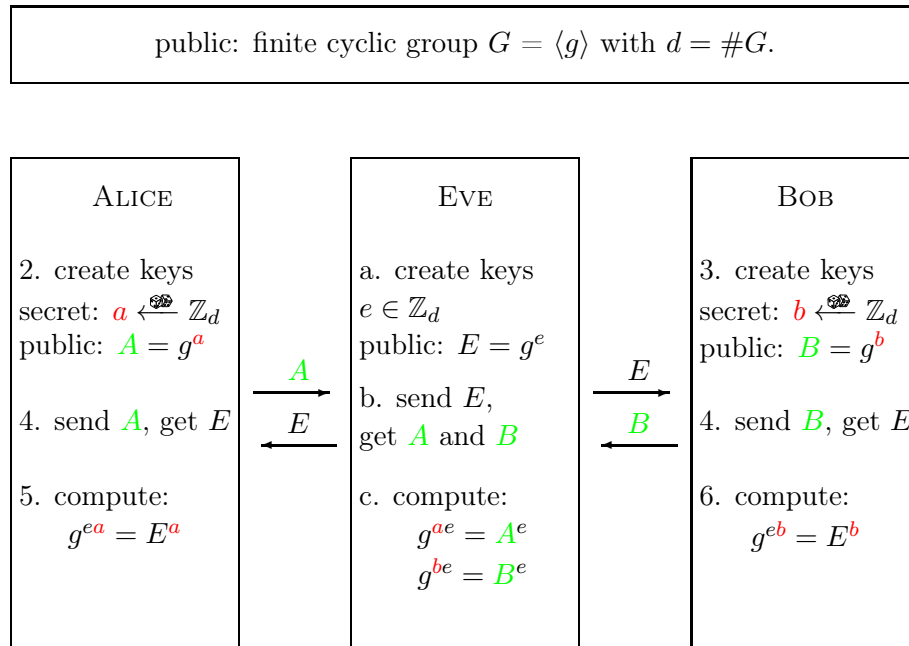


Figure 3: Woman-in-the-middle attack on the Diffie-Hellman key exchange.