1 public-key cryptosystems

each participant has a *public key* and a *secret key*. these keys specify one-to-one functions from $\mathcal{D}$ to itself. for any participant, the public adn secret keys are a m̈atched pair; *i.e.*

$$M = S_A(P_A(M)), \tag{1}$$

for any $M \in \mathcal{D}$. we require one must keep his secret key secret, and no one is able to compute the secret key function in any practical amount of time, even one can compute the public key function efficiently. the scenario for sending a message is as follows:(suppose A wants to send a message to B)

- A obtains B's public key P_B
- A computes the *ciphertext* $C = P_B(M)$, and send C to B
- B applies his secret key S_B to retrieve the original message $S_B(C) = M$

similarly, we can implement digital signatures:

- B computes his *digital signature* $\sigma = S_B M^{'}$
- B sends pair $(M^{'}, \sigma)$ to A
- A can verify that this message is from B by verify the equation $M^{'} = P_A(\sigma)$

2 the RSA cryptosystem

one creates one's public and secret keys with the following procedure:

- selete two large prime numbers p and q at random
- conpute $n = pq$
- selete a small odd integer e that is prime to $\phi(n) = (p-1)(q-1)$
- compute d as the inverse of e, modulo $\phi(n)$
- publish $P = (e, n)$ as the **RSApublickey**.
- keep pair $S = (d, n)$ as the **RSAsecretkey**.

and the function that public key specifies is $P(M) = M^e \bmod n$, and secret key, $S(C) = C^d \bmod n$. the correctness of RSA is proved from Fremat's Theorem, and the security of RSA rests on the difficulty of factoring large integers, although this is not proven.