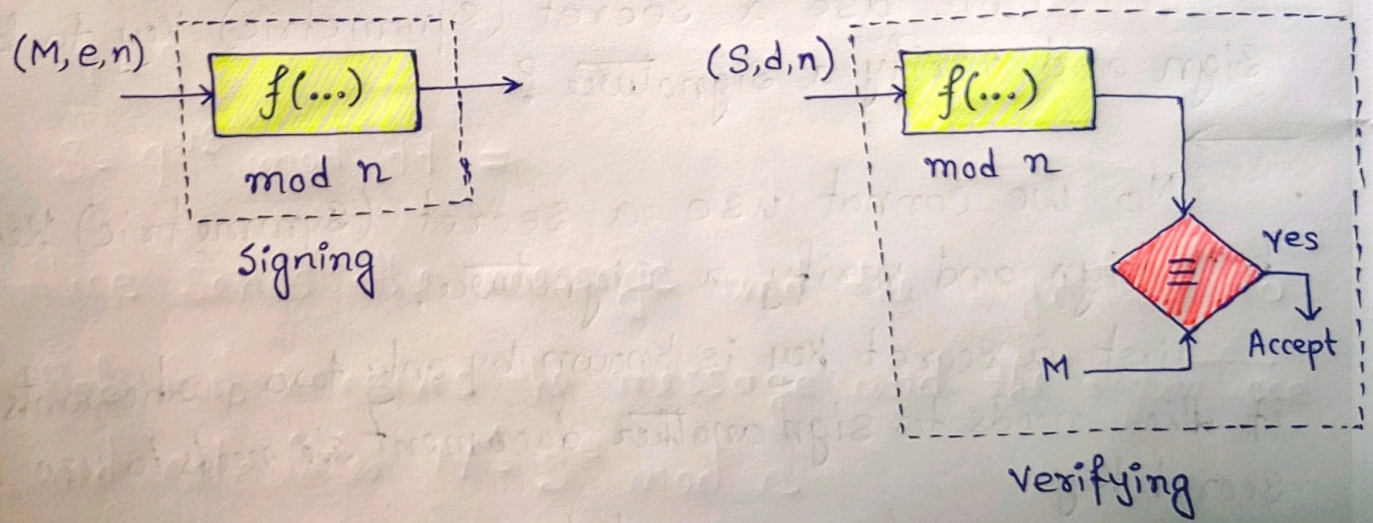
RSA Digital Signature Scheme :

The RSA idea can also be used for signing and verifying a message. In this case it is called the RSA digital signature scheme.

First, the private and public key of the sender are used.

Second, the sender uses her own private keys to sign the document; the receiver uses the sender's public key to verify it.

If we compare the scheme with the conventional way of signing, we see that the private key plays the role of the sender's own signature, the sender's public key plays the role of the copy of the signature that is available to the public.



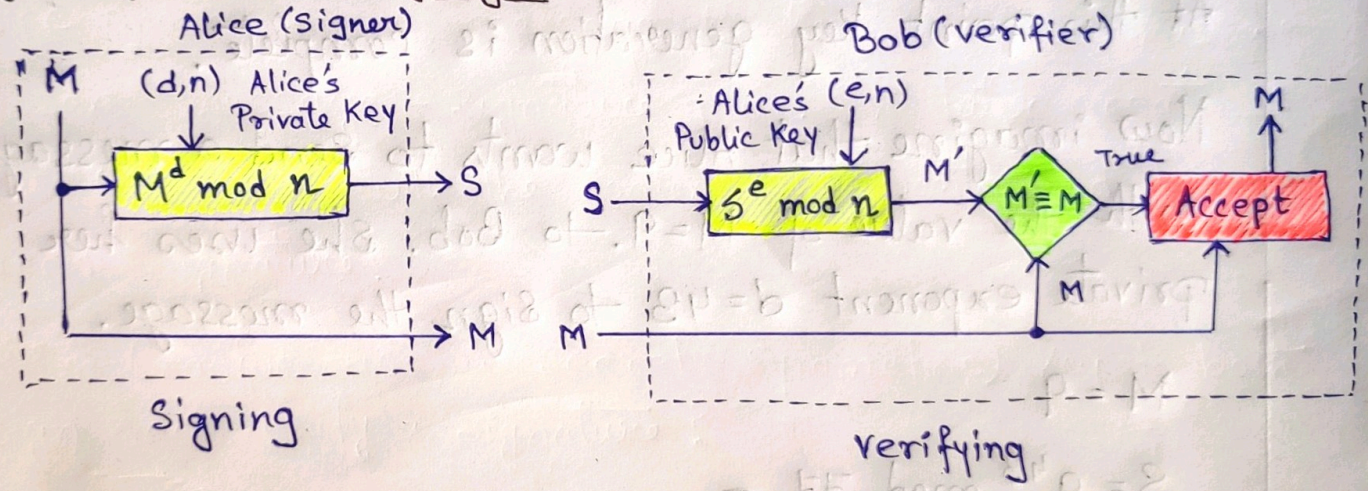
The signing and verifying sites use the same function, but with different parameters. The verifier

Compare the message and the output of the function for congruence. If the result is true, the message is accepted.

Key Generation:

Key generation in the RSA digital signature scheme is exactly the same as key generation in the RSA cryptosystem. Alice chooses two prime numbers P and q and calculate $n = P \times q$. Then Alice calculate $\phi(n) = (P-1)(q-1)$. She then chooses e, the public exponent, and calculates d, the private exponent such that $ed \equiv 1 \bmod \phi(n)$ Alice keeps d and she publicaly announces n and e.

Signing and Verifying:



Signing: Alice creates a signature out of the message using her private exponent d. $S = M^d \bmod n$ and sends the message and the signature to Bob.

Verifying: Bob receives M and S. Bob applies Alice's public exponent e to the signature to create a copy of the message $M' = S^e \bmod n$. Bob compares the value of M' and M. If two values are congruent Bob accepts the message.

Example:

As a trivial example suppose that Alice chooses $P = 11$ and $q = 7$ and calculate $n = p \times q = 77$. Then the value of $\phi(n)$ is $(p-1) \times (q-1)$; $\phi(n) = 6 \times 10 = 60$

Now Alice chooses its public key $e = 7$.

Calculate $d = de \bmod \phi(n)$

$= 7d \bmod \phi(n)$

$= 7d \bmod 60$

$= 43$

$\therefore d = 43$

At this point key generation is complete.

Now imagine that Alice wants to send a message with the value of $M = 9$ to Bob. She uses her private exponent $d = 43$ to sign the message.

$M = 9$

$S = 9^{43} \bmod 77$

Alice sends the message and the signature to Bob, Bob receives the message and the signature and calculates $M' = S^e \bmod n$

Attacks on RSA signature:

i) Key-only Attack:

Eve has access only to Alice's public key. Eve intercepts the pair (M, S) and tries to create another message M' such that $M' \equiv S^e \bmod n$. This problem is as difficult to solve as the discrete logarithm problem. Besides this is an existential forgery and normally useless to Eve.

ii) Known-Message Attack:

Here eve uses the multiplicative property of RSA. Assume that Eve has intercepted two message-signature pairs (M_1, S_1) and (M_2, S_2) that have been created using the same private key. Now Eve can create $M = (M_1 \times M_2) \bmod n$ and $S = (S_1 \times S_2) \bmod n$ and fool Bob into believing that S is Alice's signature on the message M. This attack is sometimes reffered to as multiplicative attack.

iii) Chosen Message Attack:

This attack also uses the multiplicative property of RSA. Eve can somehow ask Alice to sign two message M_1 and M_2 for her and later creates a new message $M = M_1 \times M_2$. Eve can later claim that Alice has signed M.