# Square and multiply algorithms

**Algorithm 1:** Right-to-left

```
Input: $n,\ a,\ d$// $n \in \mathbb{Z}, n \geq 2$; $a \in \mathbb{Z}_n$;
    $d \in \mathbb{Z}_{\varphi(n)}$ has bit length $\ell_d$
Output: $a^d \bmod n$
result $= 1,\ t = a$
for $i = 0,\ i < \ell_d,\ i++$ do
    // $i$th bit of $d$ is 1
    if $d_i = 1$ then
        // mutiply by $a^{2^i}$
        result $=$ result $* t \bmod n$// $a^d = \prod_{0 \leq i < \ell_d, d_i = 1} a^{2^i}$
    // $t = a^{2^{i+1}}$
    $t = t * t \bmod n$
return result
```

**Algorithm 2:** Left-to-right

```
Input: $n,\ a,\ d$// $n \in \mathbb{Z}, n \geq 2$; $a \in \mathbb{Z}_n$;
    $d \in \mathbb{Z}_{\varphi(n)}$
Output: $a^d \bmod n$
$t = 1$
for $i = \ell_d - 1,\ i \geq 0,\ i--$ do
    $t = t * t \bmod n$
    // $i$th bit of $d$ is 1
    if $d_i = 1$ then
        $t = a * t \bmod n$
return t
```