Deliverable

#1 2015-07-19 10:00AM LEO 18, R 3

Fault-resistant exponentiation algorithm

DETAILS	

Input: x, $d = (d_{l-1}, \ldots, d_0)_2$
Output: $y = x^d$

```
R[0] ←1;  R[1] ←1 , R[2] ←x ,
for i=0 to l-1
  if d_i=0 then
    R[0] ← R[0] * R[2];  (fake multiplication)
    R[2] ← R[2]^2 ;
  if d_i=1 then
    R[1] ← R[1] * R[2];
    R[2] ← R[2]^2 ;
end
return R[1]
```

Input: x, $d = (d_{l-1}, \ldots, d_0)_2$
Output: $y = x^d$

```
R[0] ←1;  R[1] ←1 , R[2] ← x ,
for i=0 to l-1
  if d_i=0 then
    R[0] ← R[0] * R[2];  (SPA protection )
    R[2] ← R[2]^2 ;
  if d_i=1 then
    R[1] ← R[1] * R[2];
    R[2] ← R[2]^2 ;
end
if R[2] ≠ R[0] * R[1] * x  then return 'error' (FA protection )
return R[1]
```

Input: x, $d = (d_{l-1}, \ldots, d_0)_m$
Output: $y = x^d$

```
for i=0 to m-1
R[i] ←1
end
R[m] ← x
for i=0 to l-2
  if d_i=0 then
    R[0] ← R[0] * R[m];  (SPA protection)
    R[m] ← R[m]^m ;
  if d_i=1 then
    R[1] ← R[1] * R[m];
    R[m] ← R[m]^m ;
…
  if d_i=m-1 then
    R[m-1] ← R[m-1] * R[m];
    R[m] ← R[m]^m ;
```

Meeting adjourned at 02:10PM