

# One time hash based signatures

- Lamport Signature

Key gen

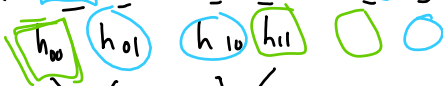
How big is private key?

$$|sk| = 2^{\lambda^2}$$

- SPHINCS



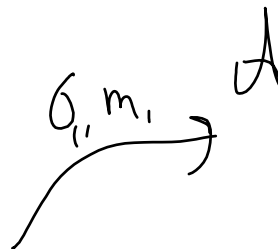
$$sk = [k_{0,0} \quad k_{0,1} \quad k_{1,0} \quad k_{1,1} \quad k_{2,0} \quad k_{2,1} \quad \dots]$$



$$pk = h_{root}$$

$$Sign(sk, 010\dots b\lambda) =$$

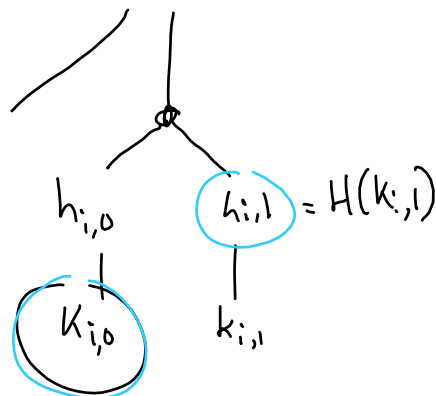
- P



- Winternitz

- Referred delegation

Lamport

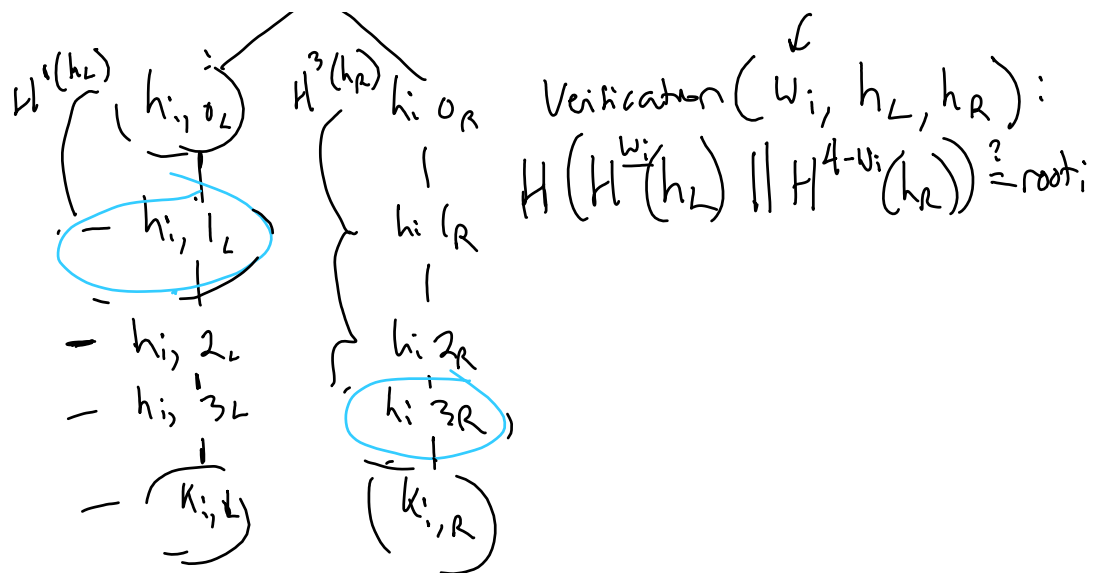


Winternitz

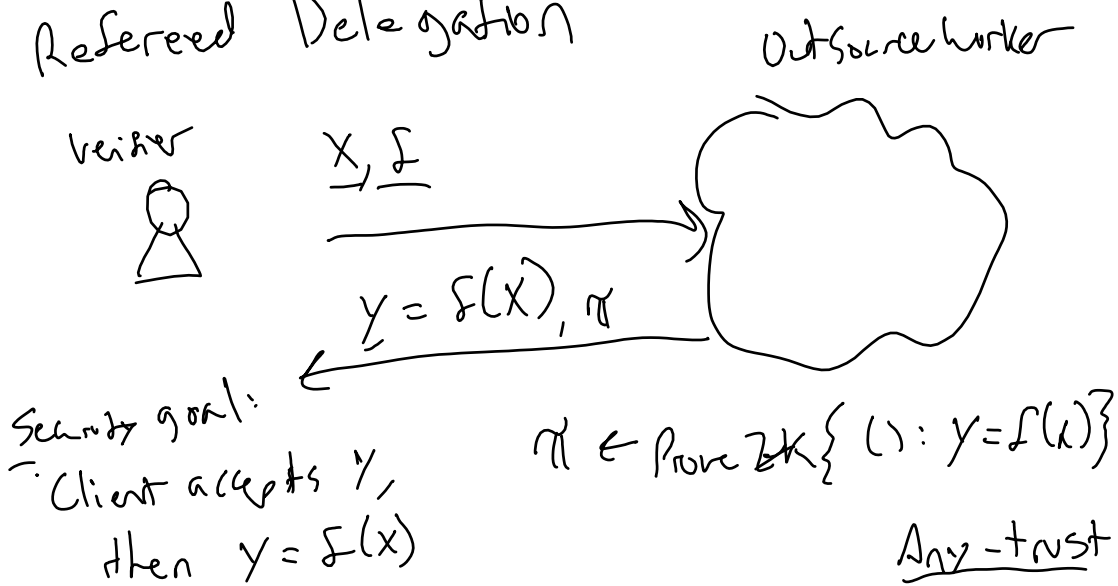
$$w_i, d_i \in \{0, 1, 2, 3\}$$

- root;

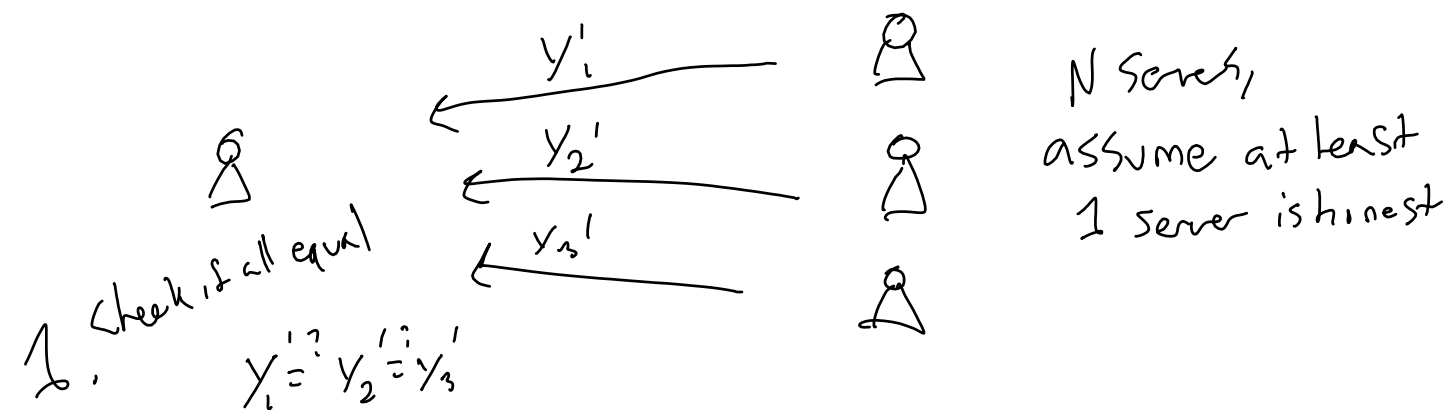
ed, 1



## Refereed Delegation



Any-trust setting



$N$  Servers,  
 assume at least  
 1 server is honest

otherwise  
 2. Run a reconciliation protocol to find a correct server.

Performance goal:

$$\text{Verifier's work} = O(n \cdot \log |f|)$$

$O(|f|)$


Idea  

$$f(x) = g(g(g(\dots g(x)))) \dots$$


$$= g^{|f|}(x)$$

Suppose  $y_1' \neq y_2'$   
 $x$



$y_1' = g^{|f|}(x)$  

$y_1'$   
 $y_2'$   $y_2' = g^{|f|}(x)$

$x$   $|f|/2$   
 $g(x)$    
 root hash  $y_1', |f|/2$   
 $y_2', |f|/2$

  
 $g(y_{1,t})$

$y_{1,t}$   $y_{1,t+1}$   
 $y_{2,t}$   $y_{2,t+1}$