Forward Secrecy of TreeKEM

Joël Alwen - Wickr

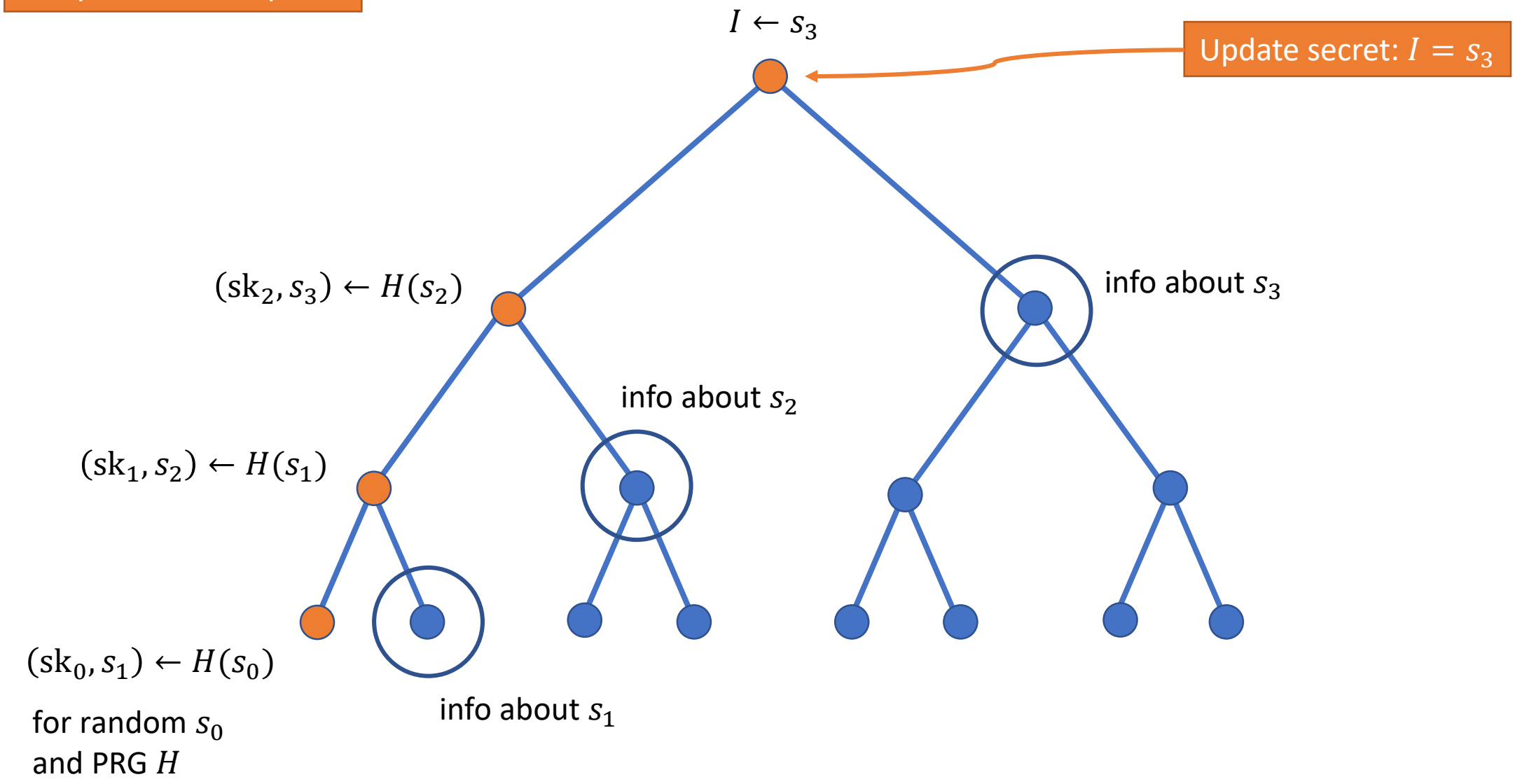
Sandro Coretti-Drayton - IOHK

Yevgeniy Dodis - NYU

Yiannis Tselekounis - NYU

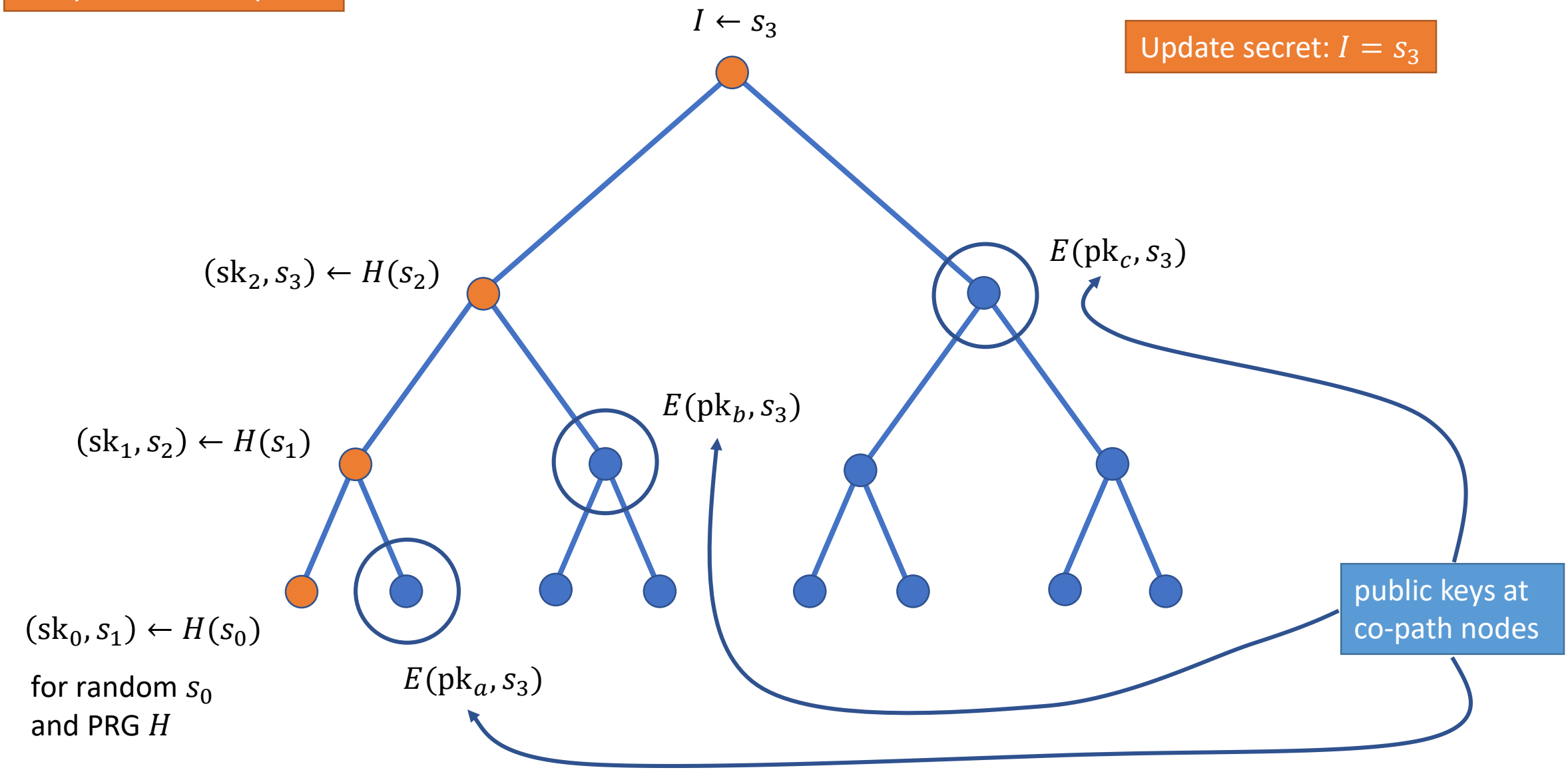
Updates in TreeKEM

Party 1 executes update

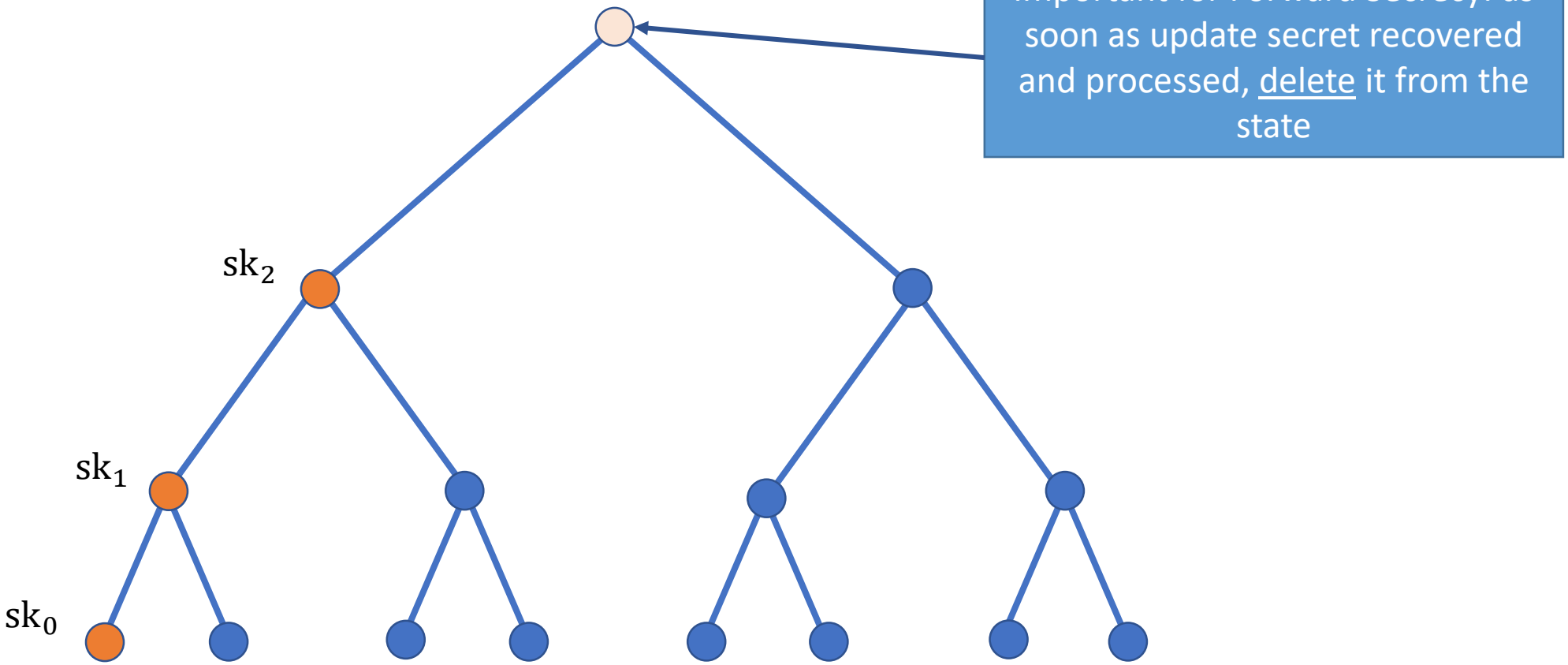


Updates in TreeKEM

Party 1 executes update



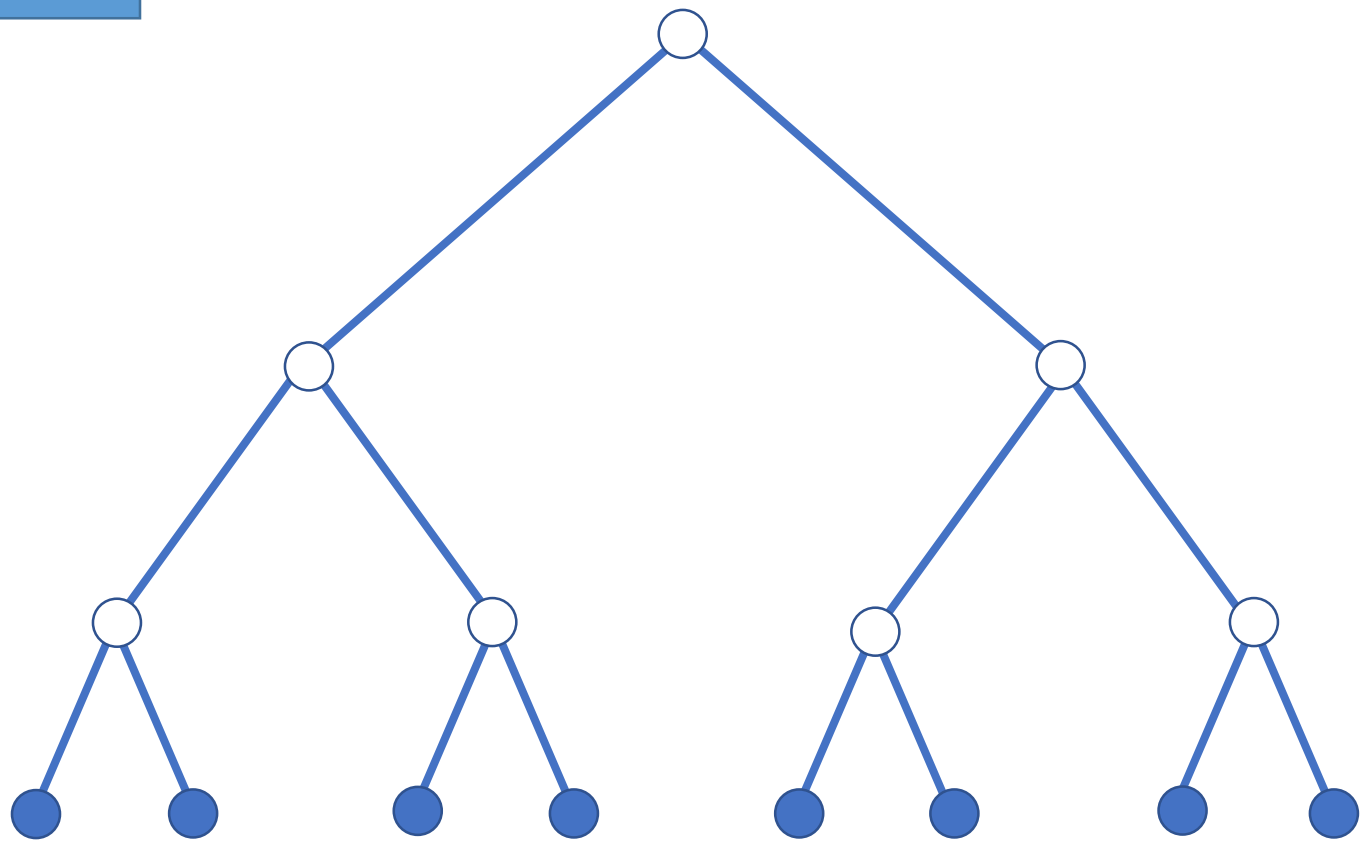
After Party 1's update



An example illustrating issues with TreeKEM’s Forward Secrecy

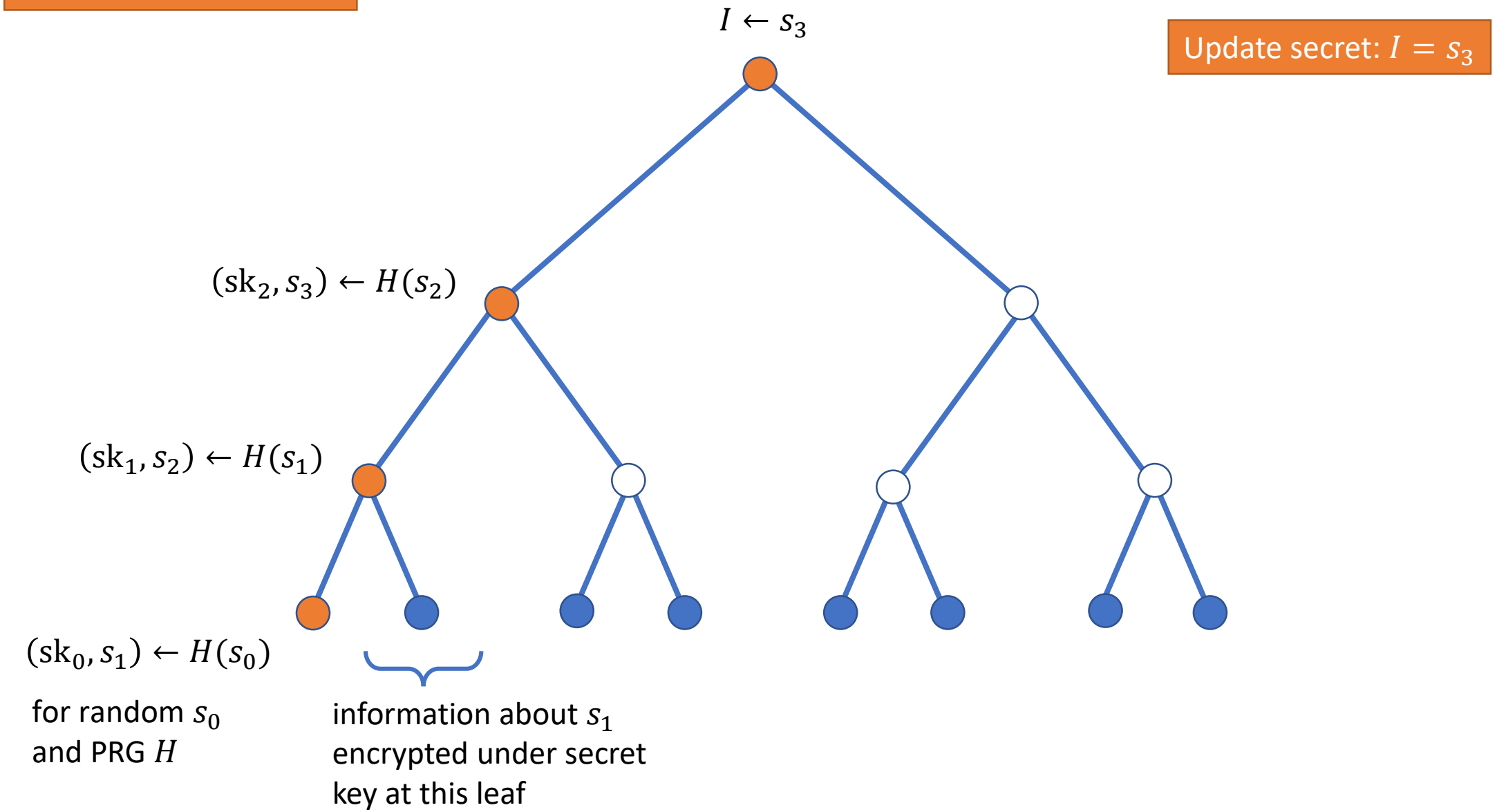
Eight group members

Internal nodes: blank initially



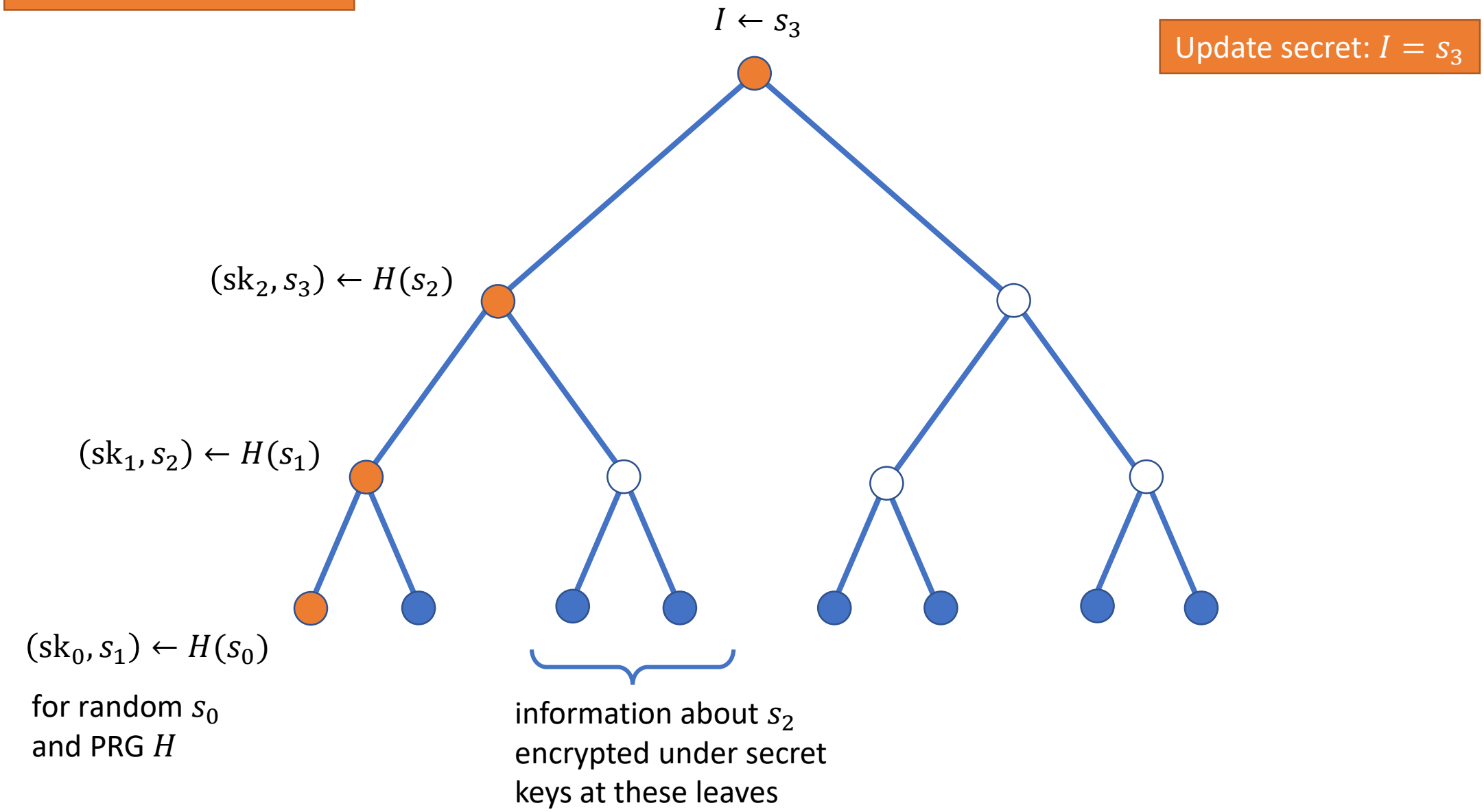
Leaves: InitKeys

Party 1 executes update



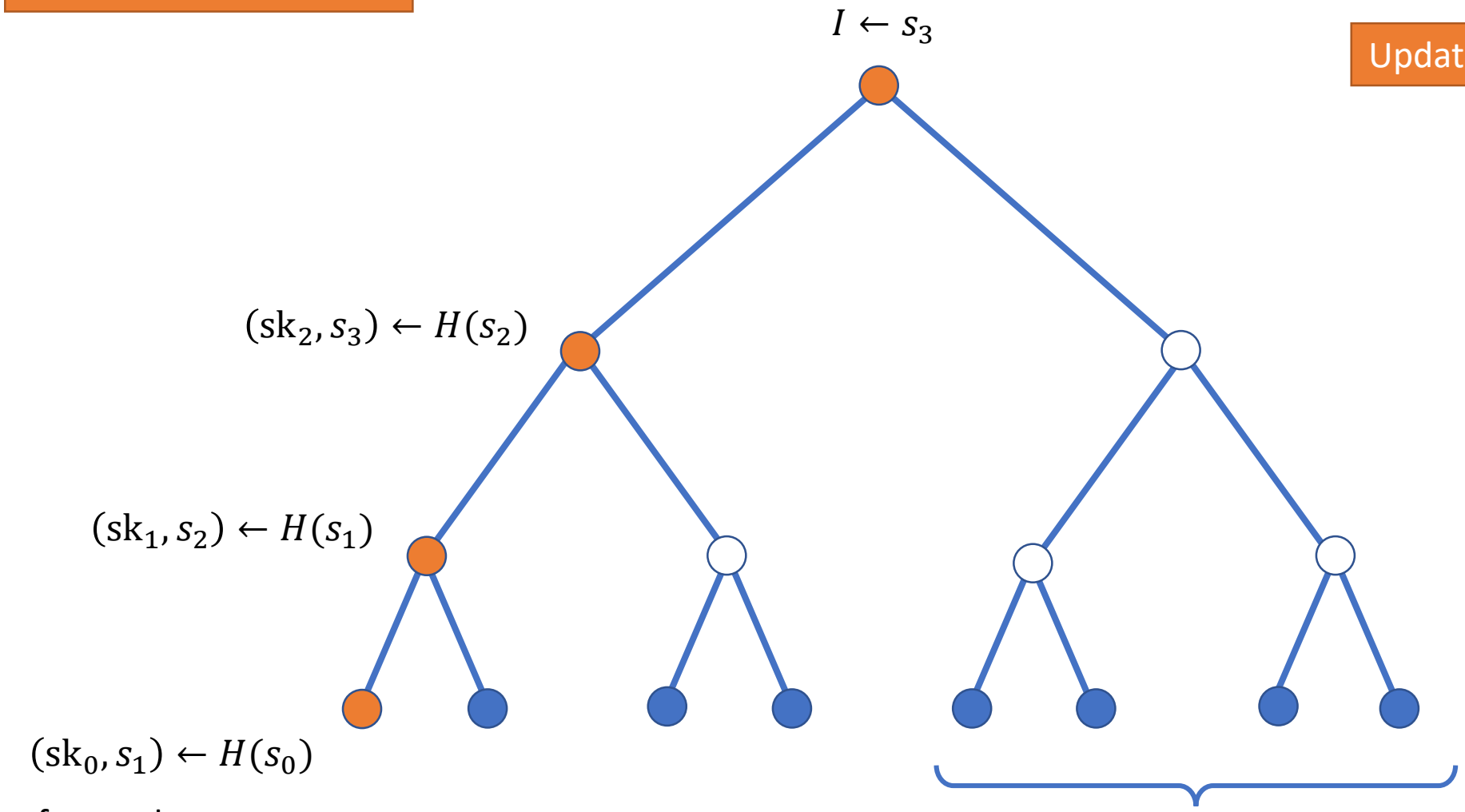
Update secret: $I = s_3$

Party 1 executes update



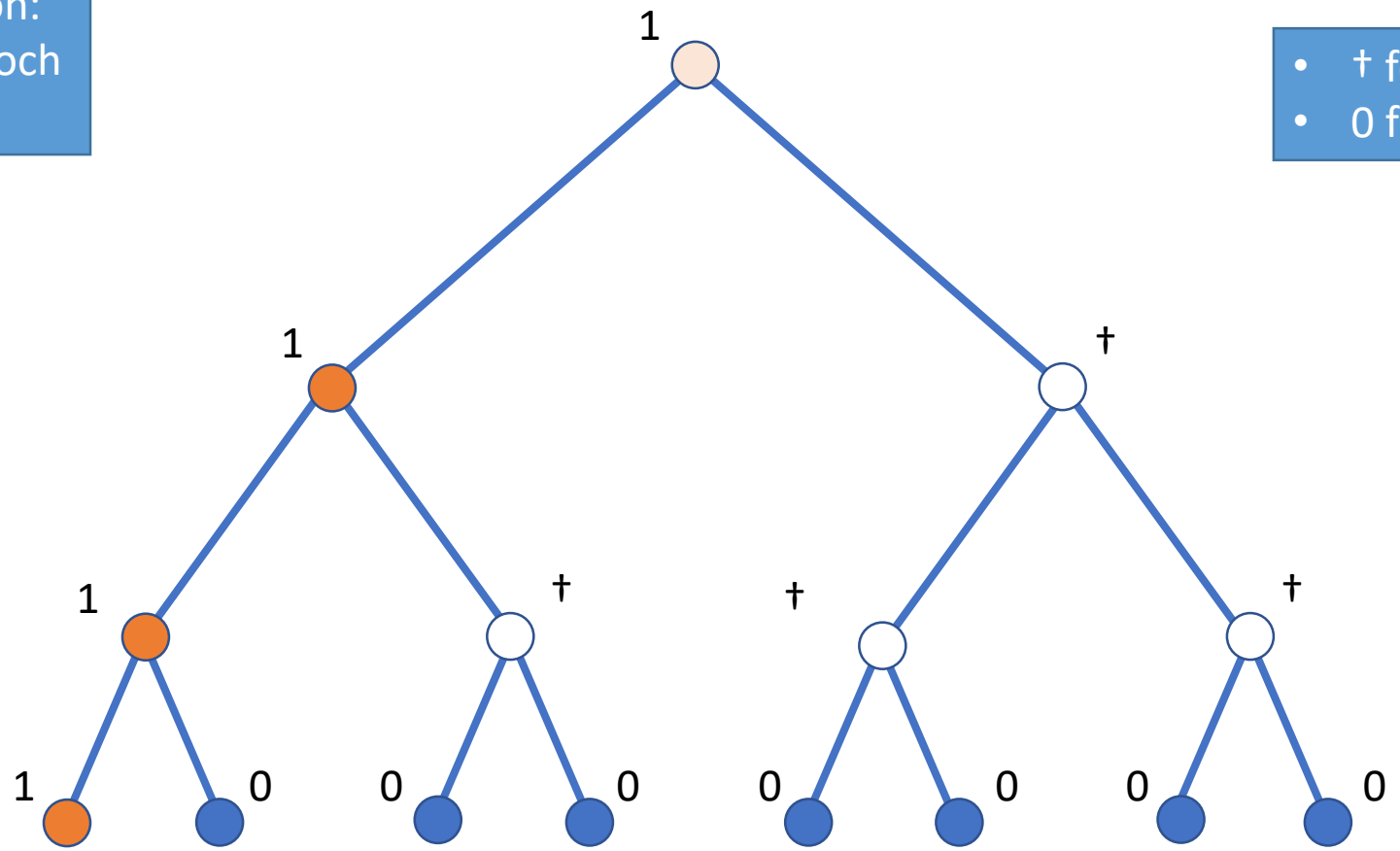
Update secret: $I = s_3$

Party 1 executes update



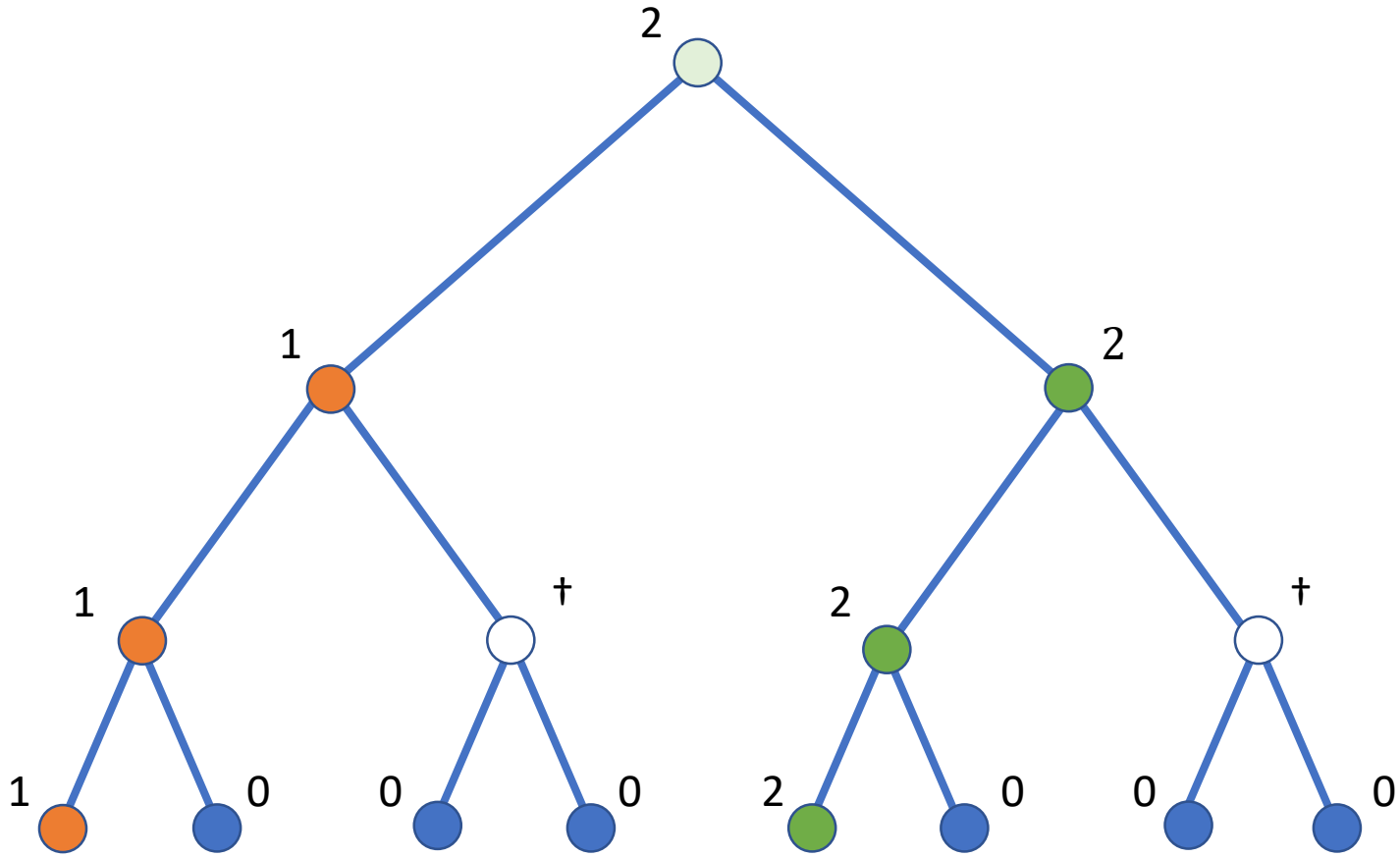
Update secret: $I = s_3$

From now on: just write epoch numbers

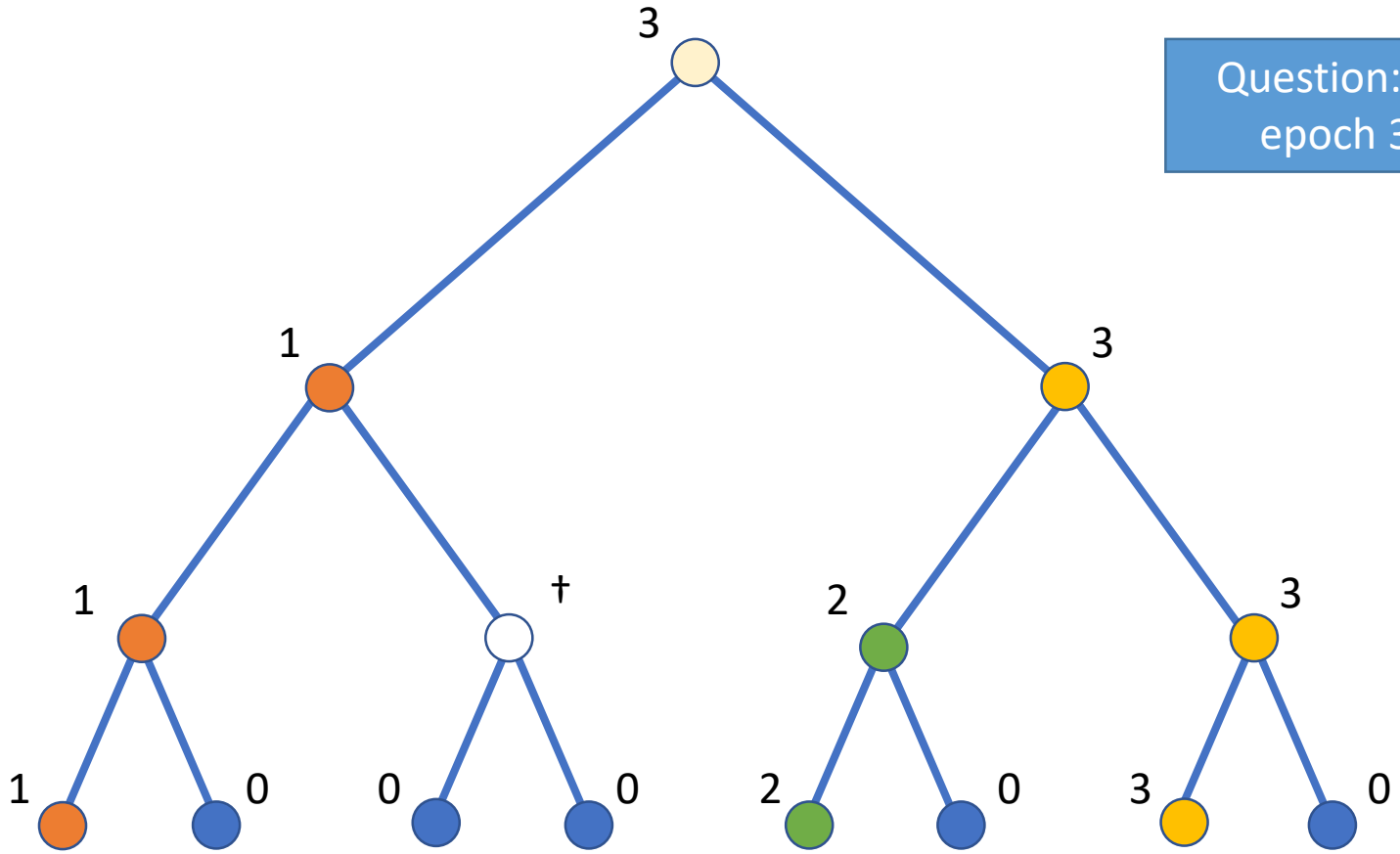


- † for blank nodes
- 0 for nodes with InitKeys

Party 5 executes update



Party 7 executes update

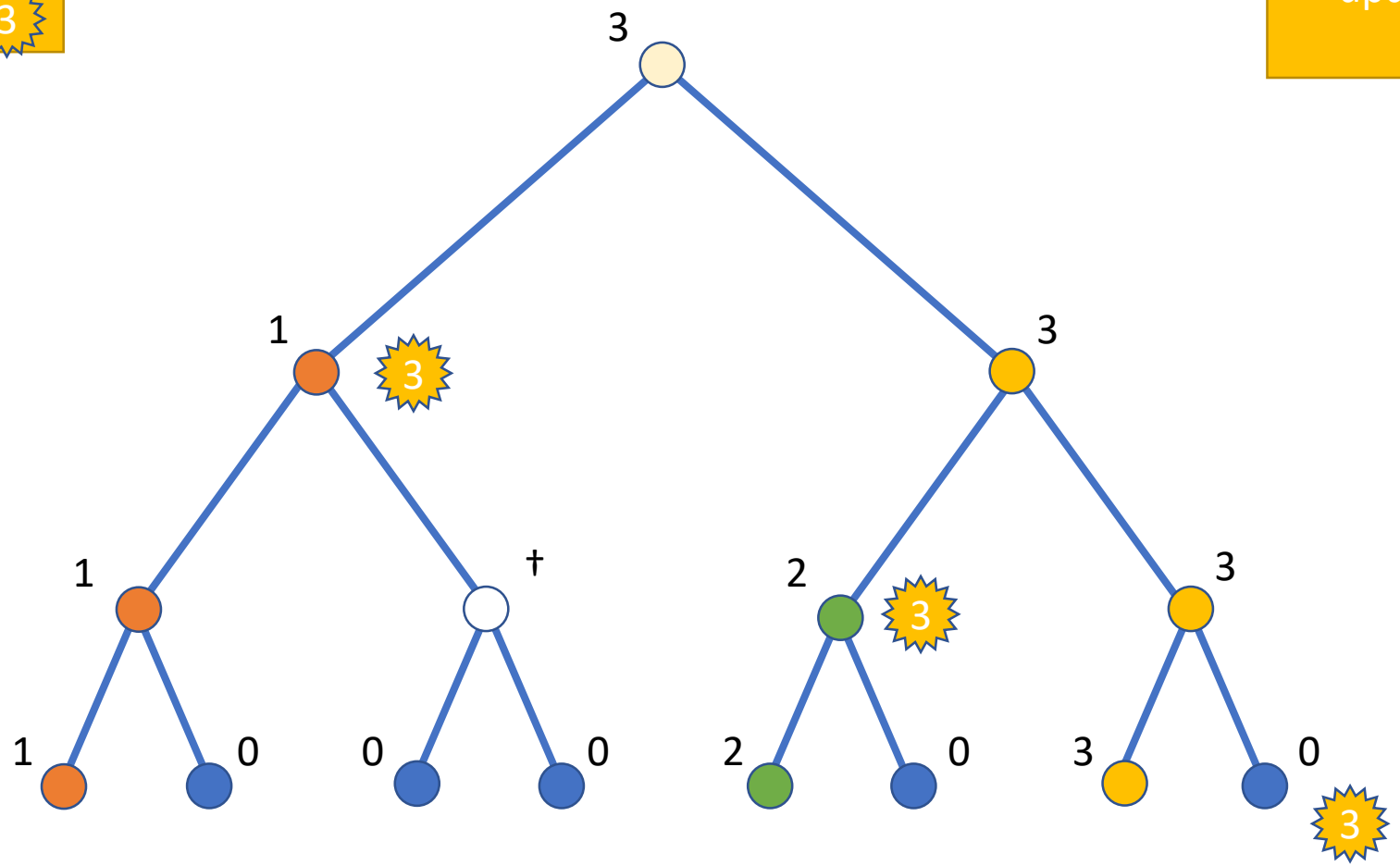


Question: Is update secret of epoch 3 forward secret?

Identifying Bombs

Information about epoch-3 update encrypted under keys of nodes with 3

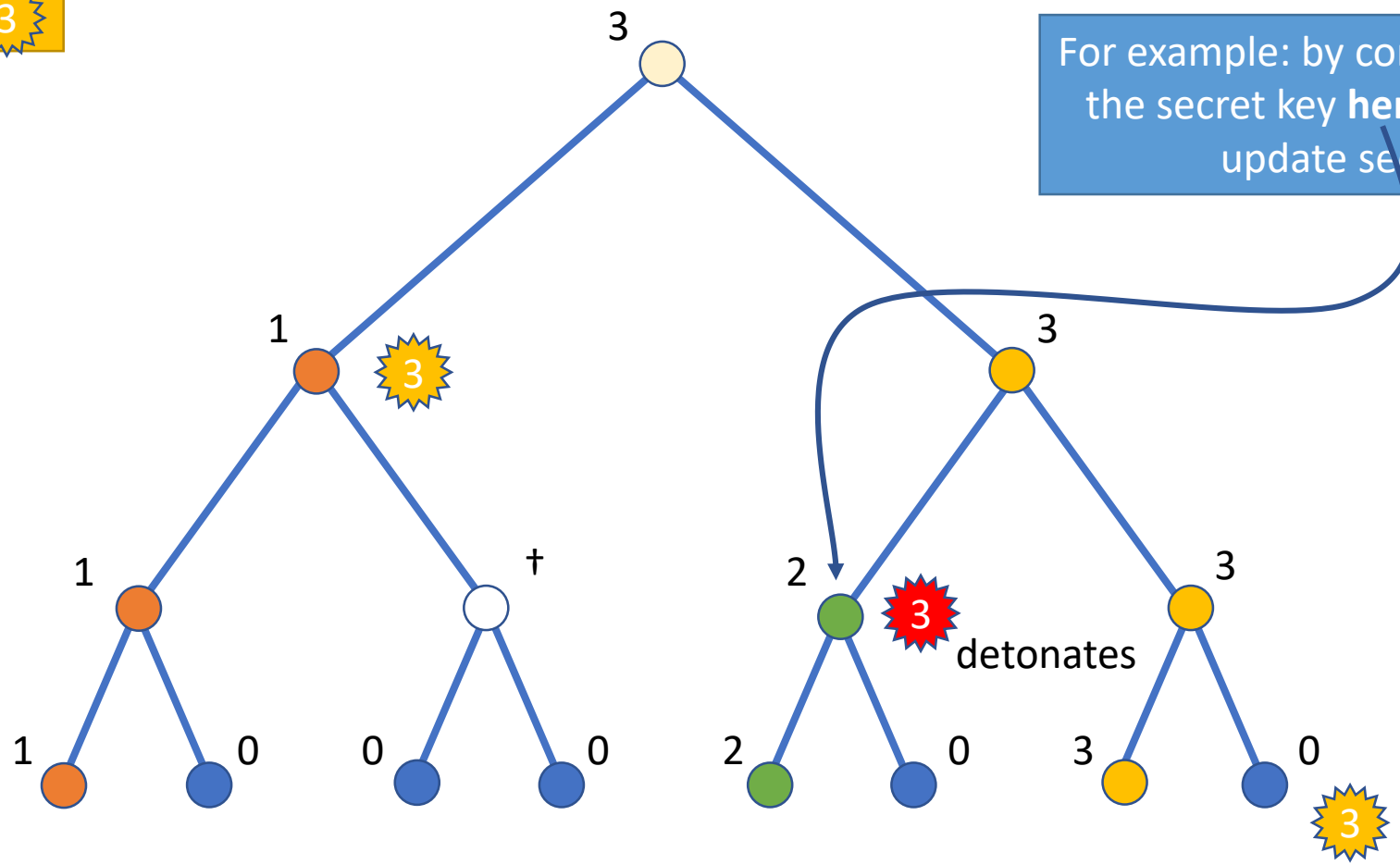
"Bomb"= Key that lets adversary recover update secret for epoch 3.



Information about epoch-3 update encrypted under keys of nodes with 3

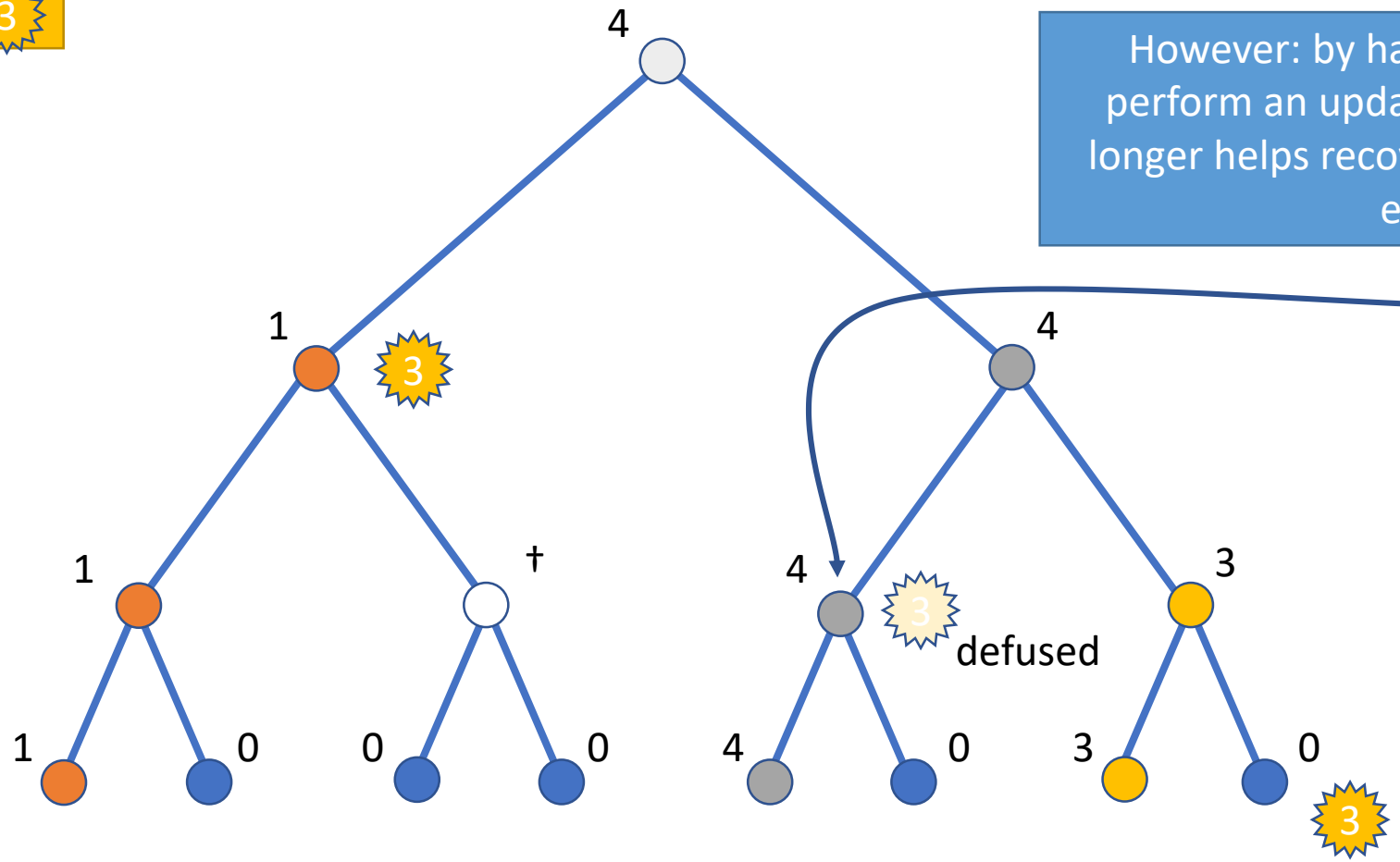
Detonating Bombs = Leaking Key

For example: by corrupting user 5 or user 6, the secret key **here** allows to recover the update secret for epoch 3



Information about epoch-3 update encrypted under keys of nodes with 3

Defusing Bombs = Refreshing Key



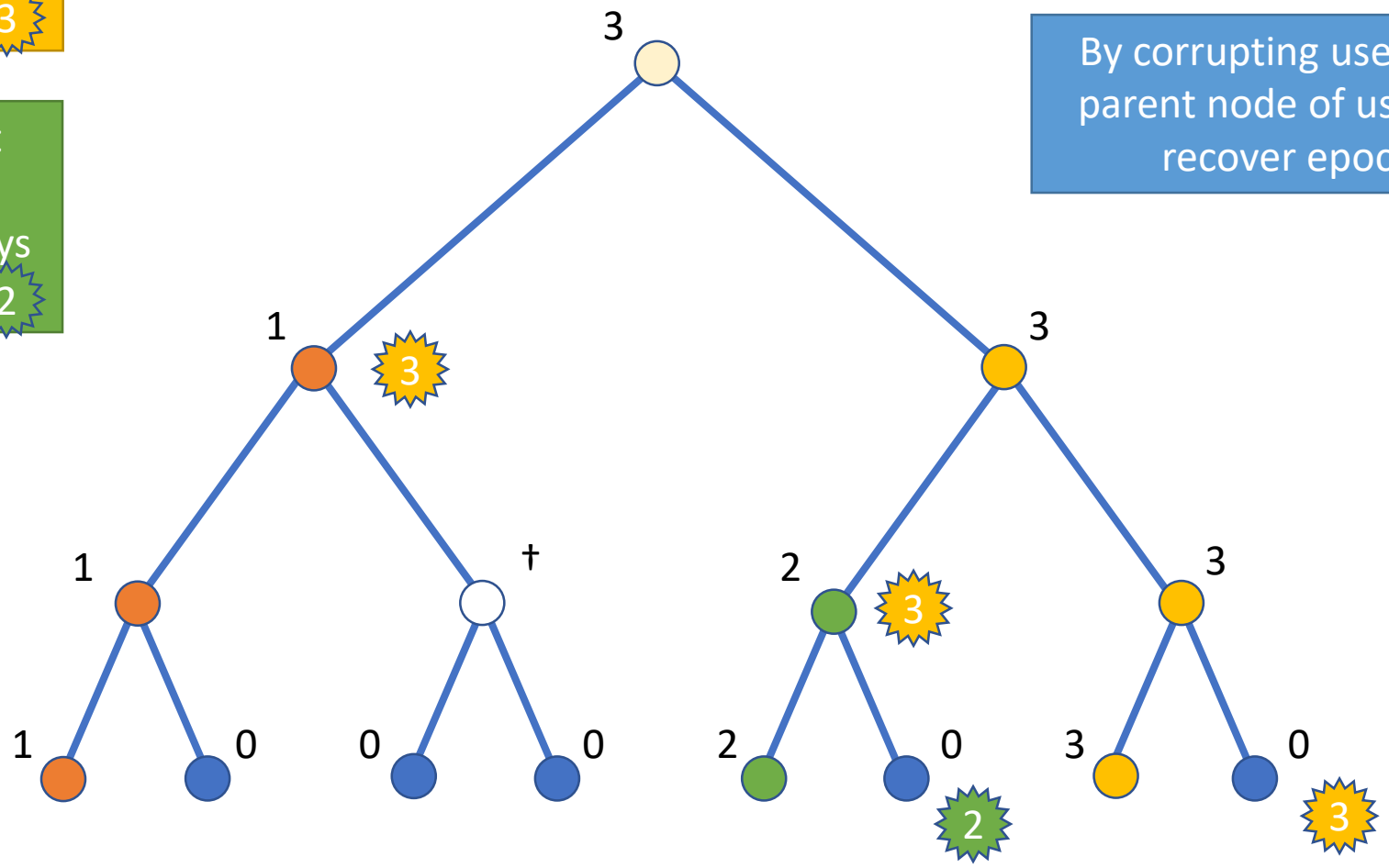
However: by having user 5 or user 6 perform an update, secret key **here** no longer helps recovering update secret of epoch 3

Information about epoch-3 update encrypted under keys of nodes with 3

Information about epoch-2 update encrypted under keys of nodes with 2

But there are more bombs!

By corrupting user 6, learn epoch-2 sk at parent node of user's leaf. With that key, recover epoch-3 update secret.



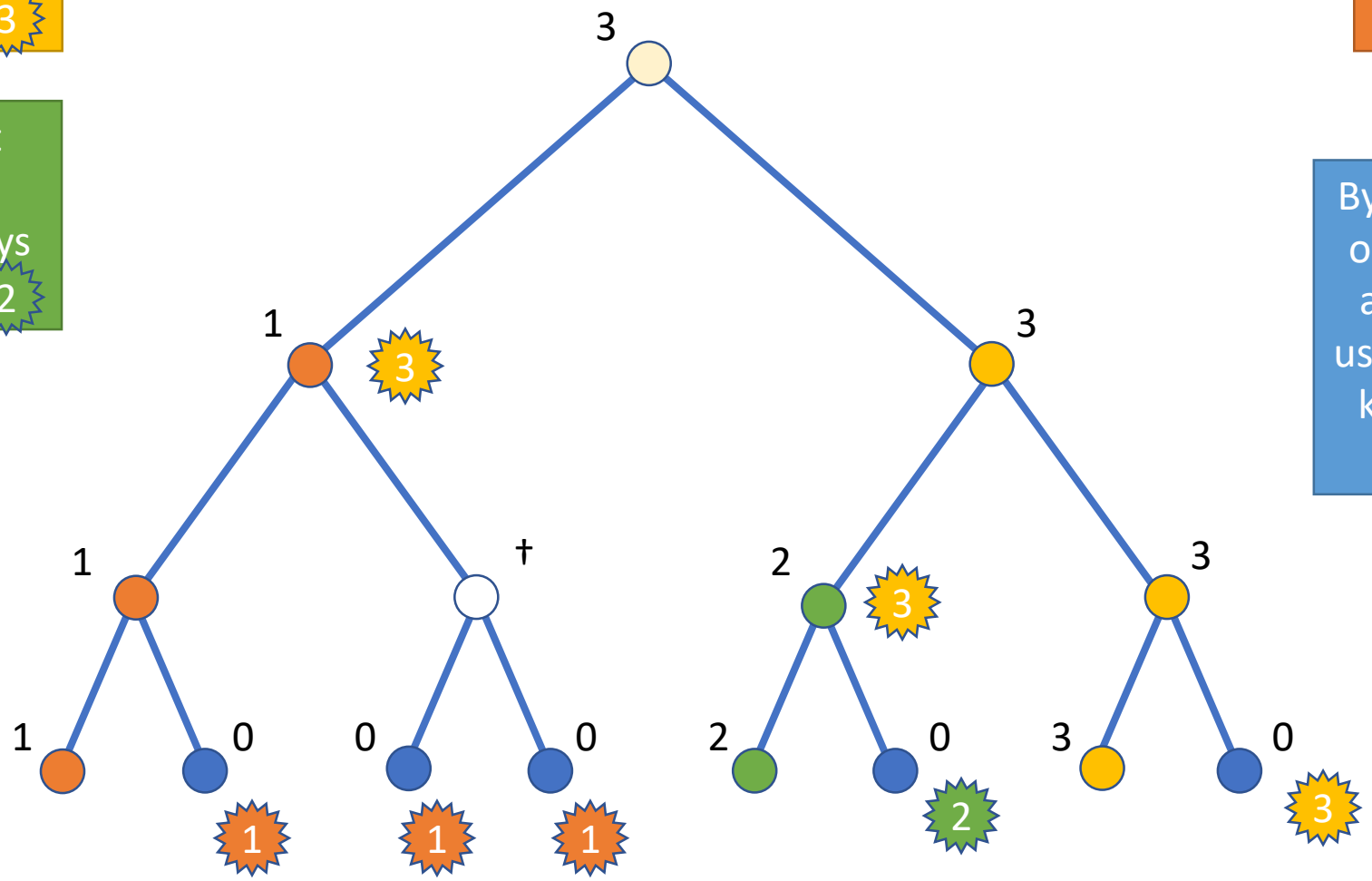
Information about epoch-3 update encrypted under keys of nodes with 3

Information about epoch-2 update encrypted under keys of nodes with 2

But there are more bombs!

Information about epoch-1 update encrypted under keys of nodes with 1

By corrupting users 2,3 or 4, learn epoch-1 sk at ancestor nodes of users' leaves. With that key, recover epoch-3 update secret.



Number of Bombs (Full Tree, No Blanks)

n_i: number of bombs with tree height i

$$n_1 = 1$$

$n > 1$: $$n_i = i + \sum_{j=1}^{i-1} n_j$$

Solves to: $n_i = 2^i - 1$

Half of the nodes in the tree have bombs!

Number of Bombs (Derivation)

n_i: number of bombs with tree height i

$$n_1 = 1$$

$$n > 1: \quad n_i = i + \sum_{j=1}^{i-1} n_j \qquad n_2 = 3$$

$$n > 2: \quad n_i - n_{i-1} = 1 + n_{i-1}$$

$$\Leftrightarrow \quad n_i = 1 + 2n_{i-1}$$

$$\Leftrightarrow \quad n_i = 1 + 2(1 + 2n_{i-2})$$

$$\Leftrightarrow \quad n_i = 1 + 2(1 + 2(1 + 2n_{i-3}))$$

$$\Leftrightarrow \quad n_i = 1 + 2 + 4 + 8n_{i-3}$$

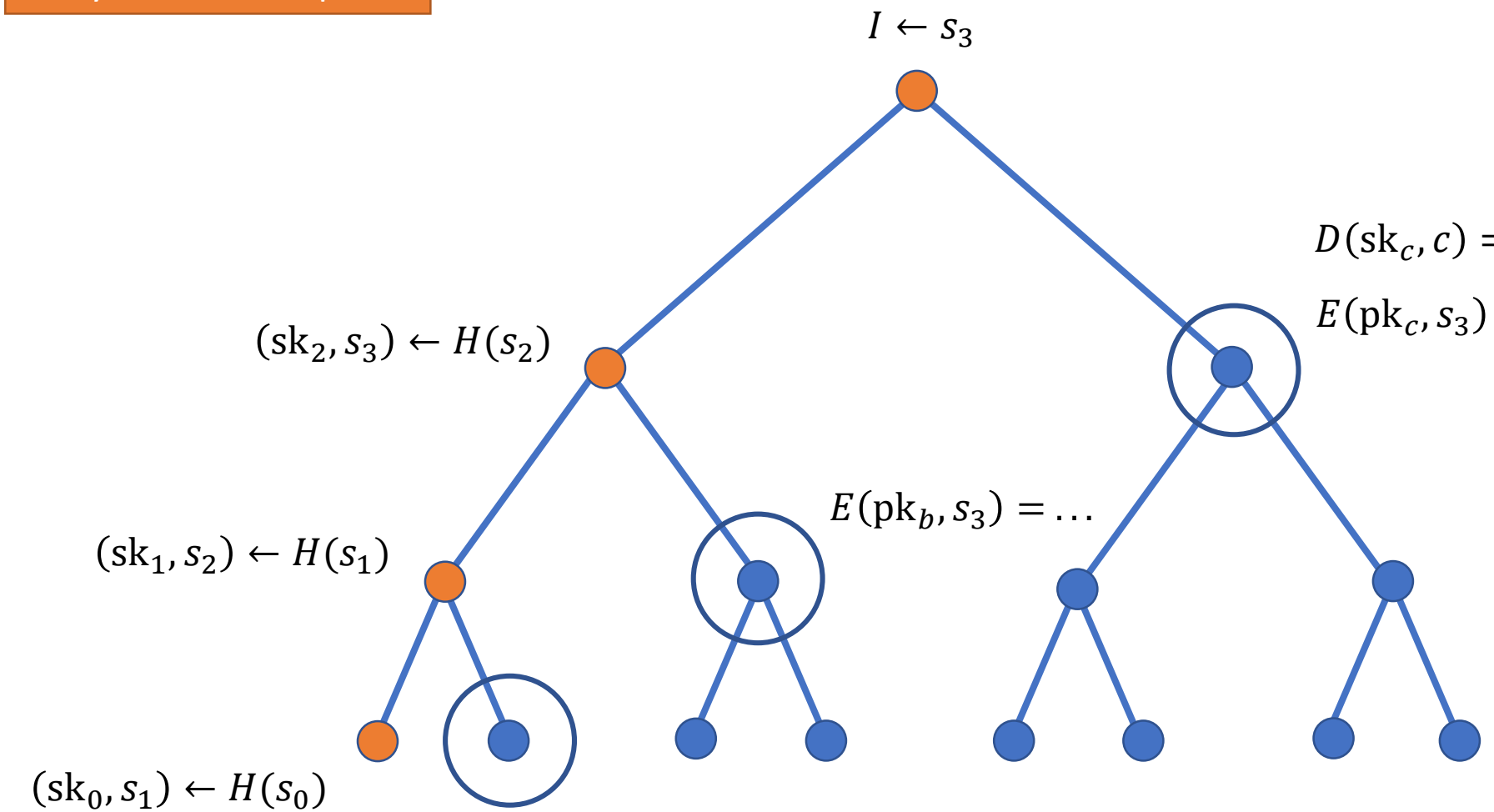
$$\Leftrightarrow \quad n_i = \sum_{k=0}^{i-3} 2^k + 2^{i-2} n_2$$

$$= \sum_{k=0}^{i-3} 2^k + 3 \cdot 2^{i-2}$$

$$= 2^{i-1} + 2 \cdot 2^{i-2} - 1 = 2^i - 1$$

Re-randomized TreeKEM (RTreeKEM)

Party 1 executes update



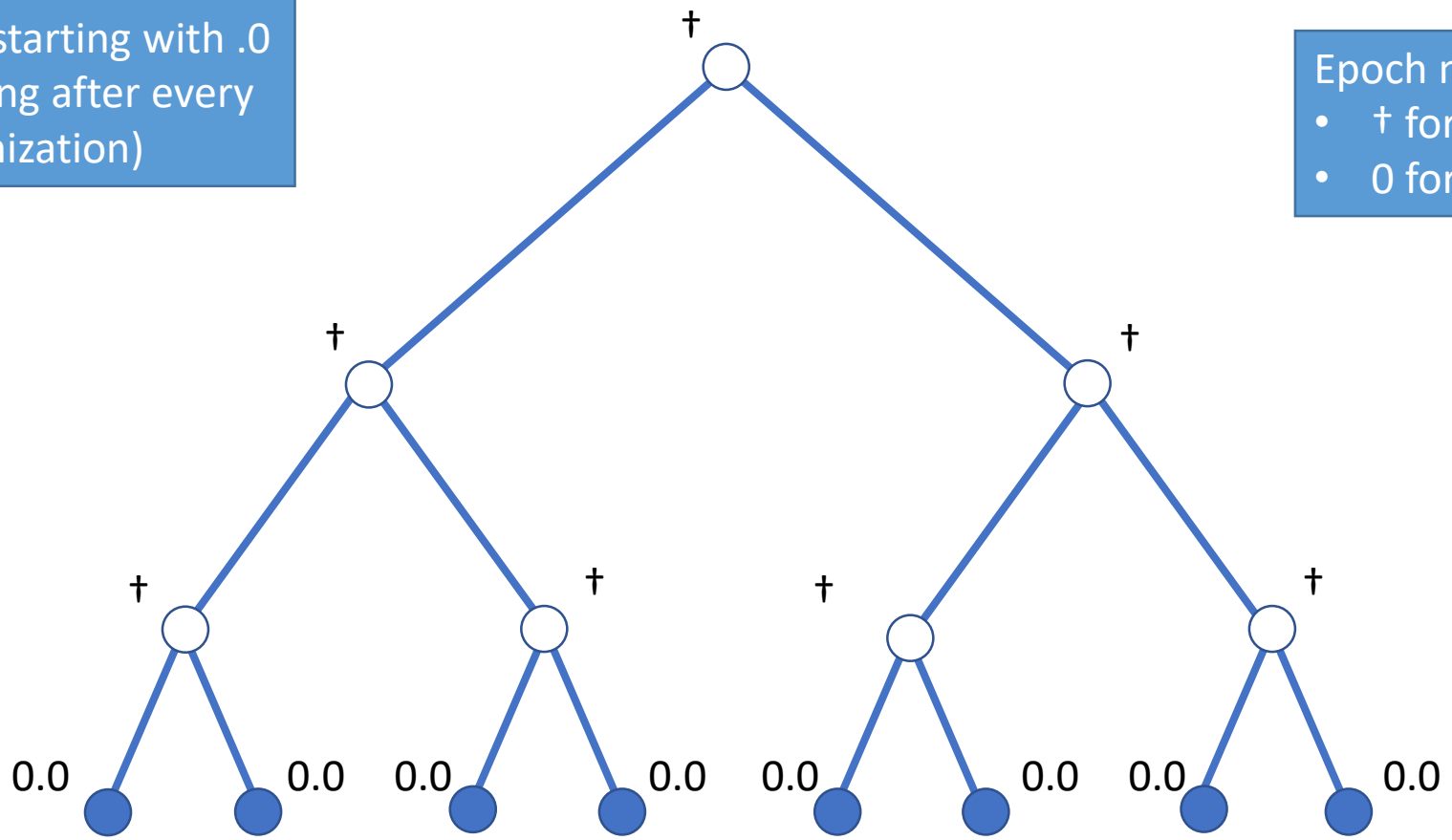
for random s_0
and PRG H

Updatable PKE: each encryption/decryption produces new pk'/sk' s.t. even given sk', messages encrypted under pk remain secret

Example with RTreeKEM

Additionally write version number of key (starting with .0 and incrementing after every rerandomization)

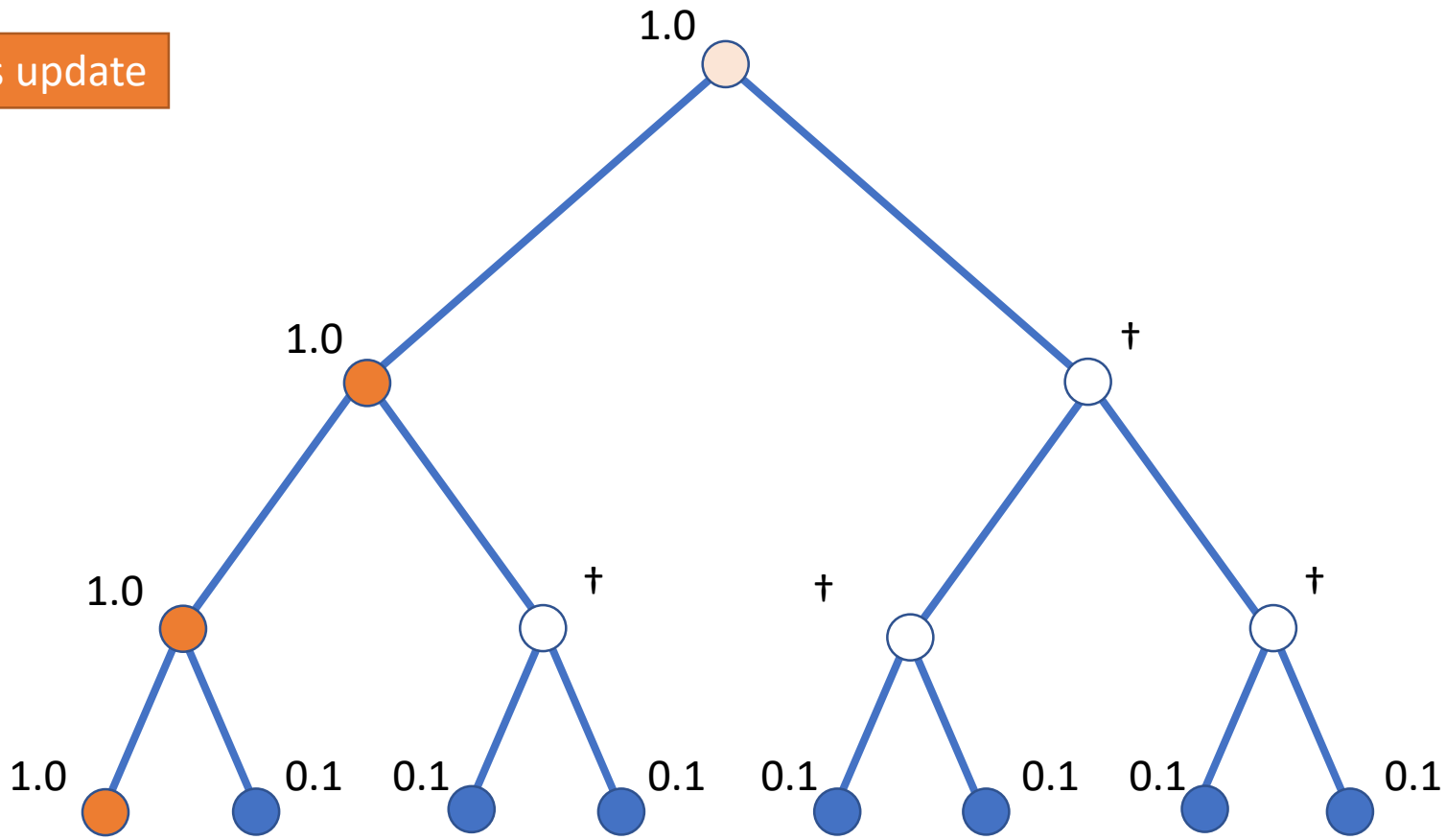
Epoch numbers (as before):
- † for blank nodes
- 0 for nodes with InitKeys



Leaves: InitKeys

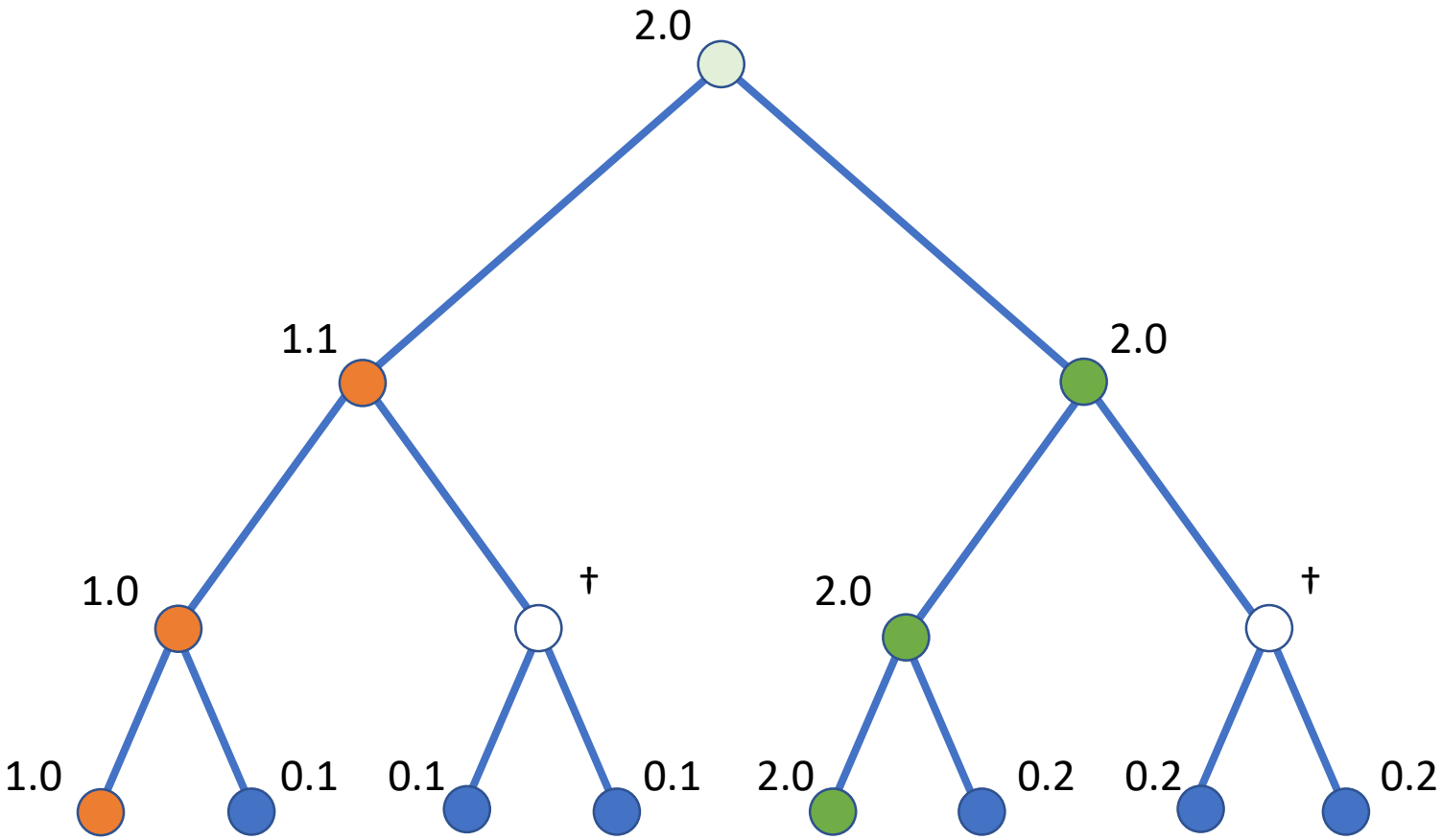
Example with RTreeKEM

Party 1 executes update



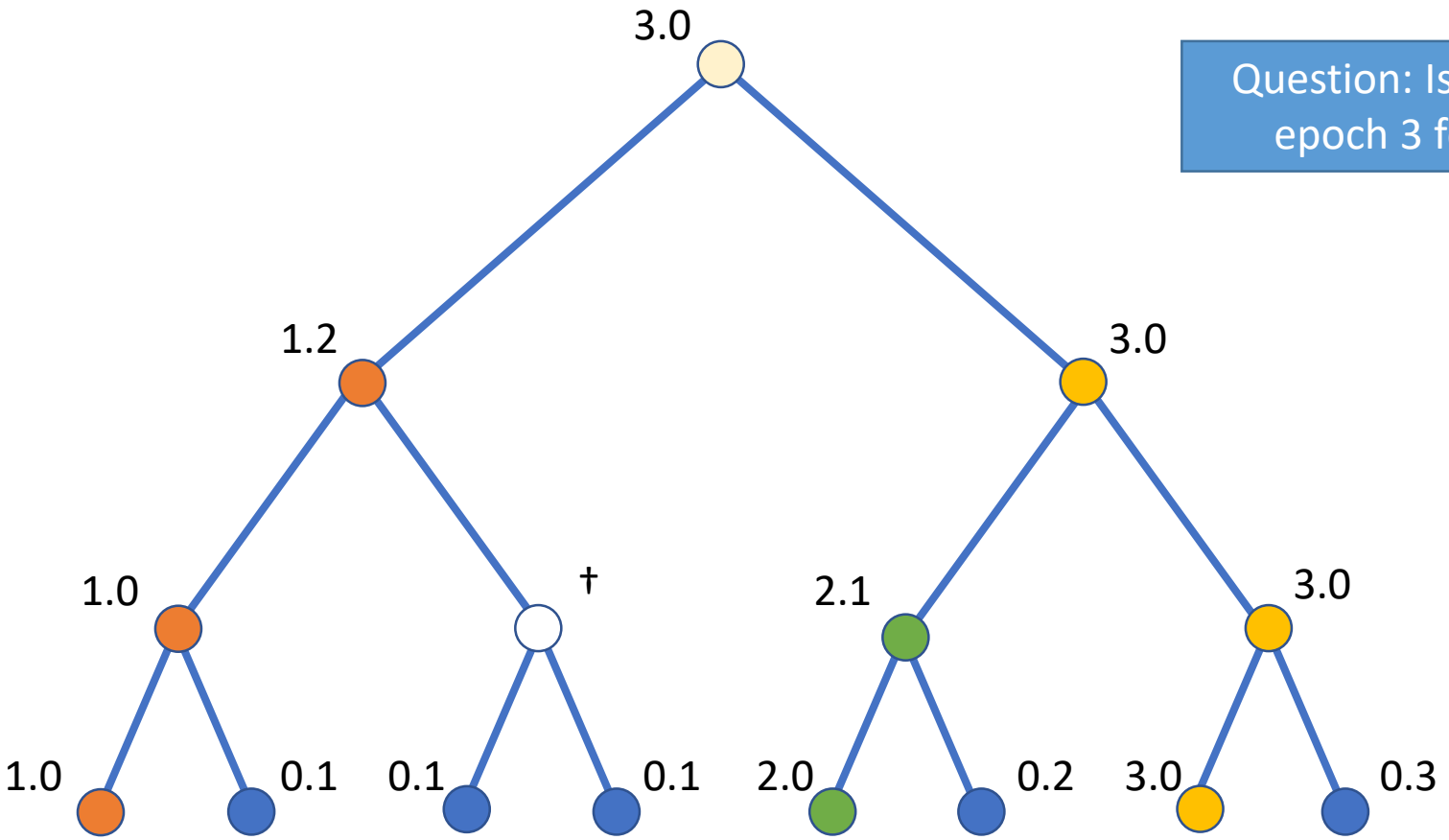
Example with RTreeKEM

Party 5 executes update



Example with RTreeKEM

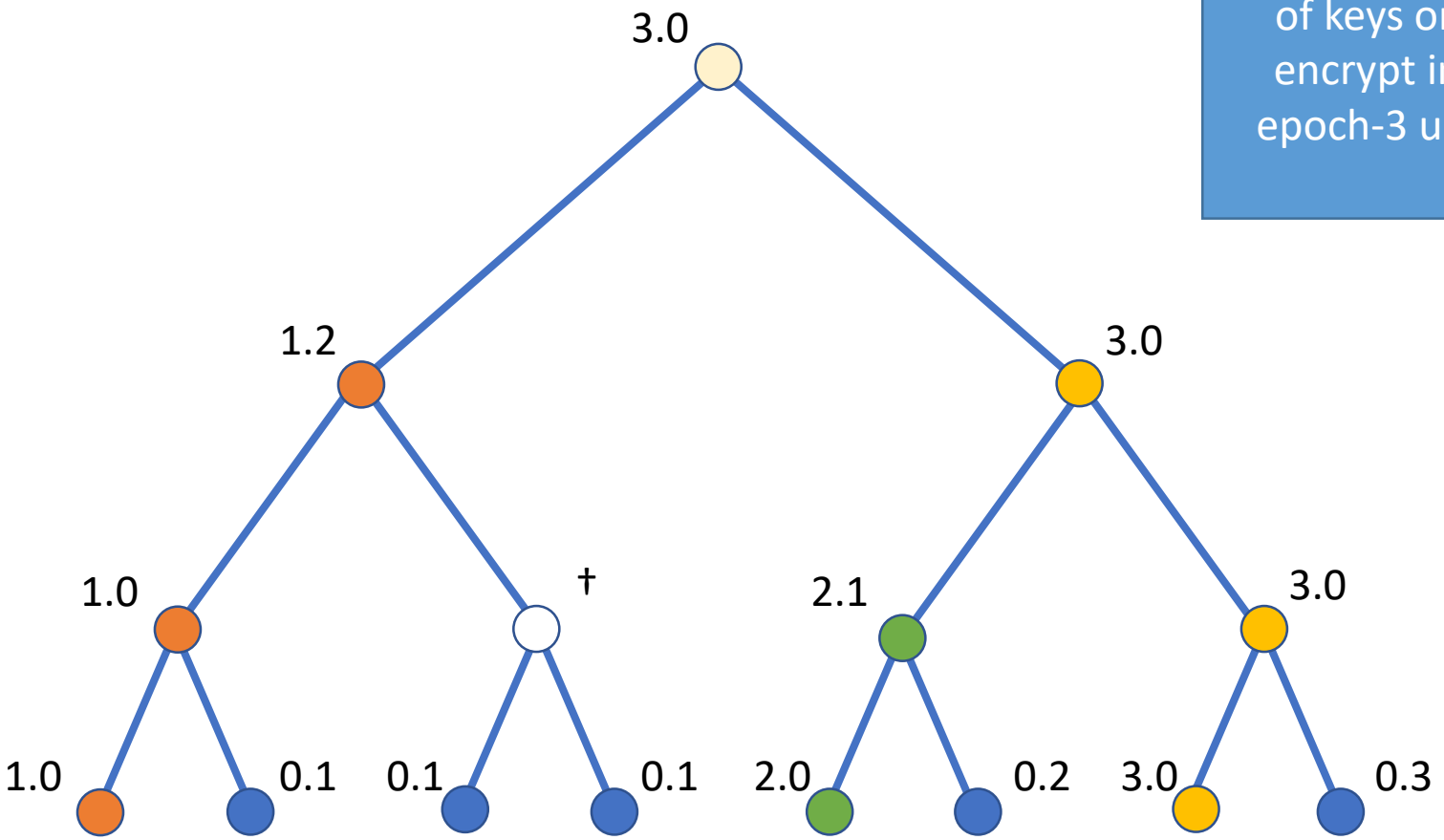
Party 7 executes update



Question: Is update secret of epoch 3 forward secret?

Example with RTreeKEM

Party 7 executes update



Yes! Versions (0.2,2.0, and 1.1) of keys on co-path used to encrypt information about epoch-3 update no longer in state!