Algorithm 2 Prepare state for the upcoming epoch e.

Input: $state, baseRole, peer$

$state.pk_{rcv}^{(e)}, state.sk_{rcv}^{(e)} \leftarrow$ generate receive public (pk) and secret key (sk)
$successful \leftarrow$ register this node with $baseRole$ intention at PKI
while $\neg successful$ **do**
    $successful \leftarrow$ register this node with $baseRole$ intention at PKI
end while
$cands \leftarrow$ receive sorted list of *mix candidates* (including public keys) for epoch e from PKI
$cands_{hash} \leftarrow hash(cands[i].pk \mid 1 \leq i \leq len(cands))$
$cands_{seed}, \_ \leftarrow Eval_{\mathrm{VDF}}(pp, cands_{hash})$ ➤ Public parameters pp known to each node.
PRNG $\leftarrow$ instantiate a new PRNG from seed $cands_{seed}$
for $c \leftarrow 1 \ldots state.NumCascades$ **do**
    for $m \leftarrow 1 \ldots state.NumMixesPerCascade$ **do**
        $state.CascadesMatrix^{(e)}[c][m] \leftarrow cands[\mathrm{PRNG}.Int(len(cands))]$ ➤ Duplicate draws are skipped.
    end for
end for
$isMix, isEntry, isExit \leftarrow$ determine *role* of node in upcoming epoch e
if $\neg isMix$ **then**
    if $baseRole = mix$ **then**
        $successful \leftarrow$ register this node with $client$ intention at PKI
        if $\neg successful$ **then**
            abort
        end if
    end if
    $isClient \leftarrow \top$
end if
$state.Clients^{(e)} \leftarrow$ receive list of *clients* for epoch e from PKI
if $isClient$ **then**
    $state.Peer^{(e)} \leftarrow$ find information on $peer$ in $state.Clients^{(e)}$
end if

return $isClient, isEntry, isExit$