Algorithm 1 Randomized sampling algorithm for estimating the impact of surveillance regulation.

```
procedure REACH(degree, hops, hubs)
    if hops = 0 then
        return 1                                                    ▷ With no hops, include just the seed.
    end if
    if hops = 1 then
        return degree + 1                                           ▷ With one hop, include the neighbors and seed.
    end if
    if hops = 2 then
        connectedHubs ← {}
        connectedPopulation ← 0
        for hub in hubs do
            if count(connectedHubs) = degree then
                break
            end if
            if random() < connectivity(hub) then
                connectedHubs = connectedHubs ∪ hub
                connectedPopulation+ = connectivity(hub)
            end if
        end for
        reach ← connectedPopulation · subscriberBase                ▷ Estimate two-hop connectivity through hubs.
        reach+ = (degree − count(connectedHubs)) · (degree − 1)     ▷ Estimate two-hop connectivity through individual
    subscribers.
        reach+ = degree                                             ▷ Estimate one-hop connectivity.
        reach+ = 1                                                  ▷ Include the seed.
        return min(reach, subscriberBase)
    end if
    if hops = 3 then
        connectedHubs ← {}
        connectedPopulation ← 0
        for hub in hubs do
            if count(connectedHubs) = degree then
                break
            end if
            if random() < connectivity(hub) then
                connectedHubs = connectedHubs ∪ hub
                connectedPopulation+ = connectivity(hub)
            end if
        end for
        reach ← connectedPopulation · subscriberBase · degree      ▷ Estimate three-hop connectivity through first-hop hubs.
        firstHopHubs ← connectedHubs
        remainingHubs ← hubs − firstHopHubs
        connectedPopulation ← 0
        for i in [0, degree − count(firstHopHubs)) do
            secondHopHubs ← {}
            for hub in remainingHubs do
                if random() < connectivity(hub) then
                    secondHopHubs = secondHopHubs ∪ hub
                    if hub not in connectedHubs then
                        connectedPopulation+ = connectivity(hub)
                        connectedHubs = connectedHubs ∪ hub
                    end if
                end if
                reach+ = 1                                          ▷ Add the individual one-hop node.
                reach+ = (degree − 1 − count(secondHopHubs)) · degree   ▷ Add the second- and third-hop nodes that are
    individual subscribers.
                reach+ = count(secondHopHubs)                       ▷ Add the second-hop nodes that are hubs.
            end for
            reach+ = connectedPopulation · subscriberBase · degree  ▷ Add the third-hop nodes through second-hop hubs.
        end for
        return min(reach, subscriberBase)
    end if
end procedure
```