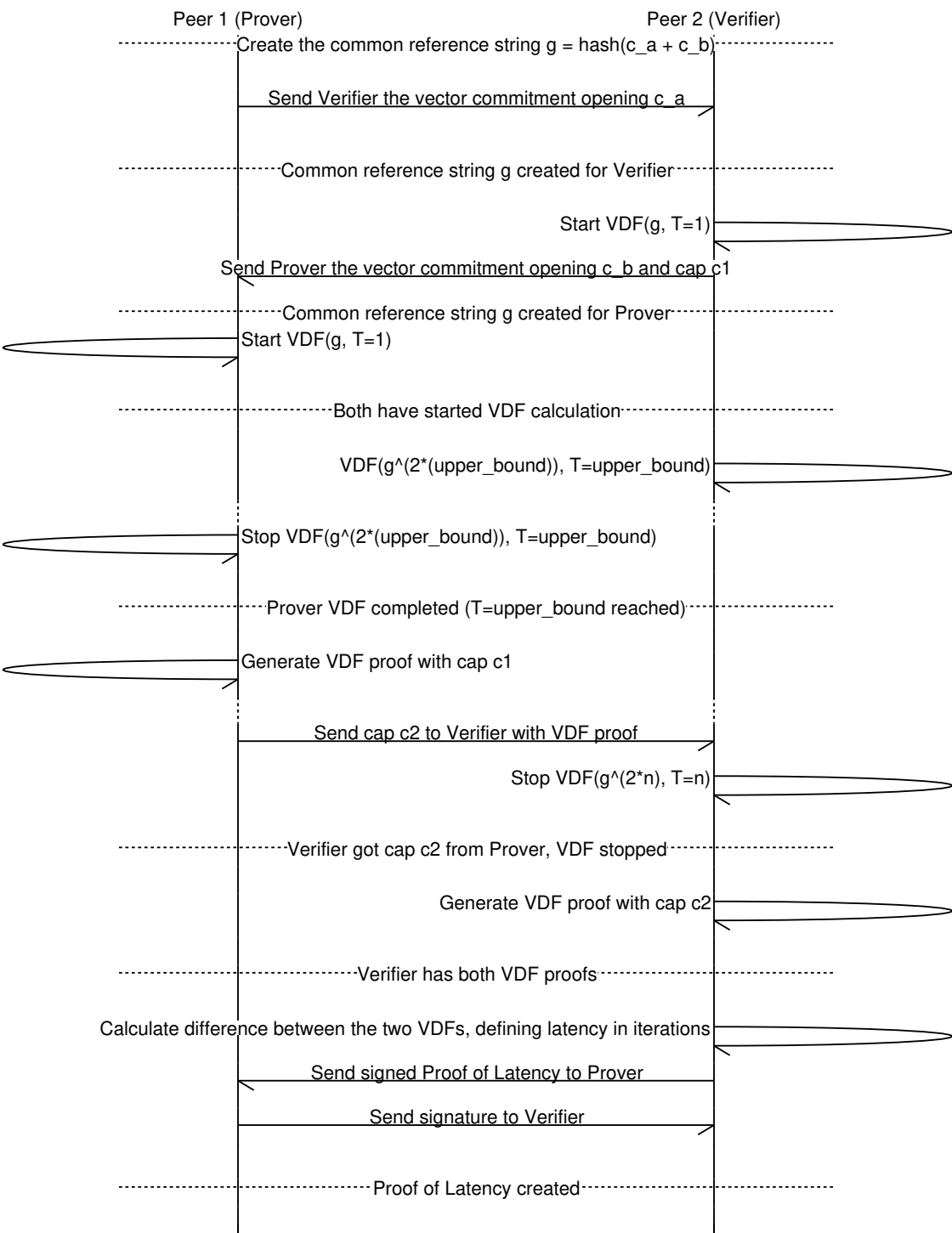
Peer 1 (Prover)
Peer 2 (Verifier)
Create the common reference string g = hash(c_a + c_b)
Send Verifier the vector commitment opening c_a
Common reference string g created for Verifier
Start VDF(g, T=1)
Send Prover the vector commitment opening c_b and cap c1
Common reference string g created for Prover
Start VDF(g, T=1)
Both have started VDF calculation
VDF(g^(2*(upper_bound)), T=upper_bound)
Stop VDF(g^(2*(upper_bound)), T=upper_bound)
Prover VDF completed (T=upper_bound reached)
Generate VDF proof with cap c1
Send cap c2 to Verifier with VDF proof
Stop VDF(g^(2*n), T=n)
Verifier got cap c2 from Prover, VDF stopped
Generate VDF proof with cap c2
Verifier has both VDF proofs
Calculate difference between the two VDFs, defining latency in iterations
Send signed Proof of Latency to Prover
Send signature to Verifier
Proof of Latency created