
# IN, LCA1: Decentralized Verifiable Computation on Distributed Ledger

Due on Spring 2017

*D.Froelicher, J.Troncoso-Pastoriza*

**Max Premi**

March 7, 2018

**Algorithm 1** Non-Interactive Range Validation

**Common Input:** $B$ the base point in the EC, $P$ the public key used to encode data, $u, l$ 2 integers and commitment $C$.

**Prover Input:** $\sigma$ the secret integer mapped to a point and $r$ a scalar in the EC such that $C = \sigma \cdot B + Pr$, $\sigma \in [0, u^l)$

**Initialization Phase**: Each server $i$ in the collective authority compute the following values :

Pick a random $x_i \in \mathbb{Z}_p$

$y_i \leftarrow Bx_i$

$A_{i,j} \leftarrow B(x_i + j)^{-1}$, $\forall j \in \mathbb{Z}_u$ and $i \in \{1, ..., m\}$

**Servers** make their signature public as well as the key $y_i$. When a query is issued by a querier $Q$, we now assume that the range are contained in the query and broadcasted by the server as usual to the data provider.

**Online Phase**: Data provider encodes the signature of the value to check in base $u$ with randomly picked $v_j$.

**for** $\forall j \in \mathbb{Z}_l$ such that $\sigma = \sum_j \sigma_j u^j$ **do**

    Picks 3 values $s_j, t_j, m_j \in \mathbb{Z}_p$

    Computes value $c = H(B, C, \sum_i y_i)$, where $H()$ is a cryptographic hash function.

    **for** $S_i$ a Server, $i \in \{0, ...m\}$ **do**

        Picks $v_j \in \mathbb{Z}_p$ and compute $V_{i,j} = A_{i,\sigma_j} v_j$

        $a_{i,j} \leftarrow e(V_{i,j}, B)(-s_j) + e(B, B)(t_j)$

    Computes $D \leftarrow \sum_j B(u^j s_j + P m_j)$

    $Z_{\sigma_j} \leftarrow s_j - \sigma_j c$ and $Z_{v_j} \leftarrow t_j - v_j c$

Finally, computes $m = \sum m_j$ and $Z_r = m - rc$

To be more precise the data provider sends the following values to ALL servers: $c, Z_r, Z_{v_j}, Z_{\sigma_j}$ with $C$ the encrypted value public, and $D, a_{i,j}$ value published to check proof.

Server $i$ checks that $D = Cc + PZ_r + \sum_j B(u^j Z_{\sigma_j})$

$a_{i,j} = e(V_{i,j}, y)c + e(V_{i,j}, B)(-Z_{\sigma_j}) + e(B, B)(Z_{v_j})$, $\forall j \in \mathbb{Z}_l$ , and publish the result.

Then the server responsible for the data provider keeps the value if all the published value match the one computed by the data provider.