SnapDealsConstraints

March 30, 2023

[16]:
```
import math
from addict import Dict as ADict
```

[11]:
```
P = 16 # Paritions
C = 1376
Pd = math.log2(P) # parition inclusion proof depth
SHA = 45331
Poseidon = {2: 367, 8: 565}
ApexLeafs = 128
d = 30 # log2(data)
```

[38]:
```
ConstraintsSHA = ADict()
ConstraintsSHA.Apex = ApexLeafs * SHA
ConstraintsSHA.PartitionInclusion = Pd * SHA
ConstraintsSHA.PerC.PathRs = 2 * d/3 * Poseidon[8] # 2x R paths
ConstraintsSHA.PerC.ApexSelection = ApexLeafs
ConstraintsSHA.PerC.PathD = (d - Pd - math.log2(ApexLeafs)) * SHA
ConstraintsSHA.ChallageGen = Poseidon[2] * math.ceil(C/P/9) # we get 9␣
↪challages per PRF call
ConstraintsSHA.PerC.RhoGen = Poseidon[2] + 7 # hselect
ConstraintsSHA.PhiGen = Poseidon[2]
ConstraintsSHA
```

[38]:
```
{'Apex': 5802368,
 'PartitionInclusion': 181324.0,
 'PerC': {'PathRs': 11300.0,
  'ApexSelection': 128,
  'PathD': 861289.0,
  'RhoGen': 374},
 'ChallageGen': 3670,
 'PhiGen': 367}
```

[43]:
```
import numbers
def is_number(x):
    return isinstance(x, numbers.Number)

totalSHAConstraints = C * sum(ConstraintsSHA.PerC.values()) + P *␣
↪sum(filter(is_number, ConstraintsSHA.values()))
```

```
totalSHAConstraints
```

[43]: 1297176880.0

[41]:
```
ConstraintsPoseidon = ADict()
ConstraintsPoseidon.PerC.PathRs = 2 * d/3 * Poseidon[8] # 2x R paths
ConstraintsPoseidon.PerC.PathD =  d/3 * Poseidon[8]
ConstraintsPoseidon.ChallageGen = Poseidon[2] * math.ceil(C/P/9) # we get 9␣
 ↪challages per PRF call
ConstraintsPoseidon.PerC.RhoGen = Poseidon[2] + 7 # hselect
ConstraintsPoseidon.PhiGen = Poseidon[2]
ConstraintsPoseidon
```

[41]:
```
{'PerC': {'PathRs': 11300.0, 'PathD': 5650.0, 'RhoGen': 374},
 'ChallageGen': 3670,
 'PhiGen': 367}
```

[44]:
```
totalPoseidonConstraints = C * sum(ConstraintsPoseidon.PerC.values()) +␣
 ↪sum(filter(is_number, ConstraintsPoseidon.values()))
totalPoseidonConstraints
```

[44]: 23841861.0

[46]:
```
Improvment = totalSHAConstraints / totalPoseidonConstraints
Improvment
```

[46]: 54.40753471383798