HEIR `mgmt` Dialect and Passes

Hongren Zheng

2025-04-17

TOC

Subsection 1

`mgmt` Dialect

mgmt Dialect

- mgmt stands for *Management*
- FHE RLWE Schemes have management operations
  - User: arithmetic IR
  - Backend: scheme IR
  - Compiler's duty to insert them
- Home for them in a scheme-agnostic way
  - mgmt.modreduce: Rescaling or Modulus Switching
  - mgmt.relinearize: Relinearization
  - mgmt.bootstrap: Bootstrapping

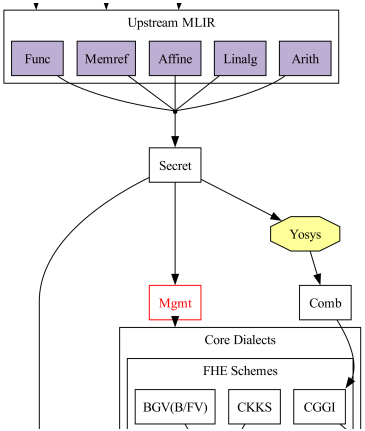
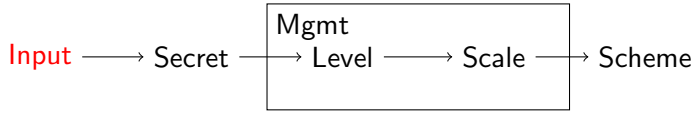
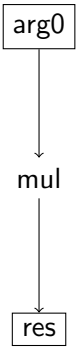


Figure: Diagram from heir.dev

Example: User Input

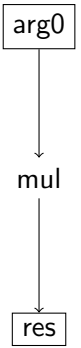



```
func @main(%arg0 : f32) {
  %res = arith.mulf %arg0, %arg0
  return %res
}
```

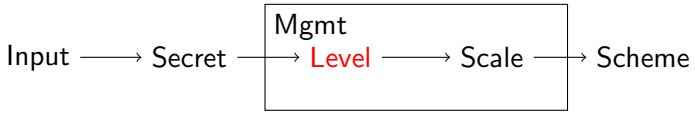


Example: Secret-Arithmetic IR

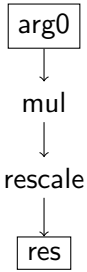

```
func @main(%arg0 : !secret<f32>) {
  secret.generic {
    %res = arith.mulf %arg0, %arg0
    return %res
  }
}
```



Example: Level Management

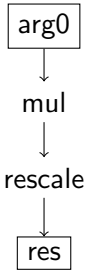



```
func @main(%arg0 {level = 1}) {
  secret.generic {
    %mul {level = 1} = arith.mulf %arg0, %arg0
    %res {level = 0} = mgmt.modreduce %mul
    return %res
  }
}
```



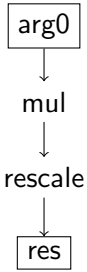
Example: Scale Management


```
func @main(%arg0 {scale = 45}) {
  secret.generic {
    %mul {scale = 90} = arith.mulf %arg0, %arg0
    %res {scale = 45} = mgmt.modreduce %mul
    return %res
  }
}
```



Example: Scheme


```
func @main(%arg0: !lwe.ct) {
  %mul = ckks.mul %arg0, %arg0
  %res = ckks.rescale %mul
  return %res
}
```



Subsection 2

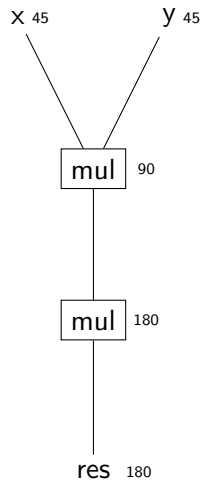
mgmt passes

Level Management: `secret-insert-mgmt`

- OpenFHE `SetMultiplicativeDepth`: User specifying level
- `secret-insert-mgmt-<scheme>`
  - Compiler can compute it!
  - Insert level management ops
  - Annotate `level`
  - Get max `level` for parameter generation
- `<scheme>`: Different schemes have different policy
  - e.g. B/FV has no level management
- This pass also does other management like `relinearize`
- Big TODO: Bootstrapping placement
  - We do have the op `mgmt.bootstrap`
  - We do not have a good placement policy implemented

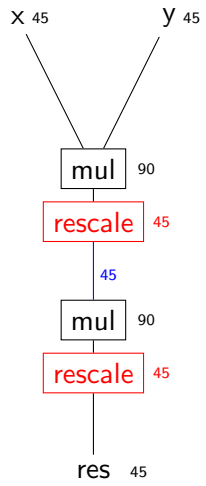
Three styles of rescaling

- Example: $(xy)^2$
- Problem: Scale blow up
- Need rescaling!
- Three styles of rescaling placement
  - After mul
  - Before mul
  - Before mul including the first mul



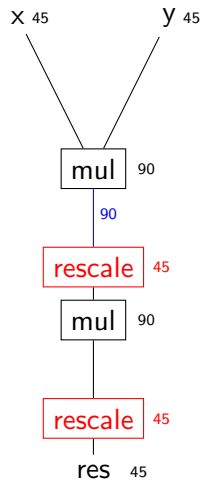
Rescaling: After Mul

- "Textbook way": CKKS paper
  - After multiplication, insert rescale
  - to control the scale
  - (or noise for BGV)
- Between two multiplications
  - The scale is small (45)



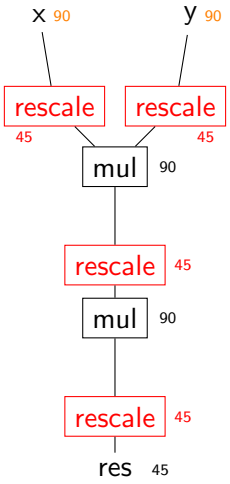
Rescaling: Before Mul

- OpenFHE: FLEXIBLEAUTO
- Look like the same? No
- Between two multiplications
  - Scale is big (90)
- Benefit: smaller noise growth
  - If we have operations in blue region
  - e.g. rotation, addition
  - Their noise could be *hided*
  - By the rescale below
- HEIR defaults to this



Rescaling: Before Mul including the first mul

- OpenFHE: FLEXIBLEAUTOEXT
- One step further
- Benefit: Even smaller noise
  - Encryption noise reduced
- Penalty
  - One more level: 3 levels
  - Encrypt at double degree (90)

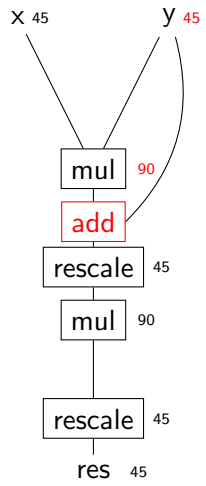


Scale Management: `populate-scale`

- Why this is a standalone pass:
- *secret-insert-mgmt* only inserts mgmt operations
- We do not have concrete scale value at that time
- Only after *generate-param* pass
  - Ask user for default scale
  - Relies on *secret-insert-mgmt*
- The scale is known now: populate to all ciphertext
- Also handles cross-level operations

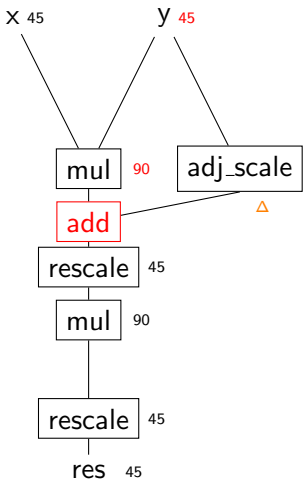
Example: cross-level add

- Example: $(xy + y)^2$
- Addition: scale mismatch
  - xy: scale 90
  - y: scale 45



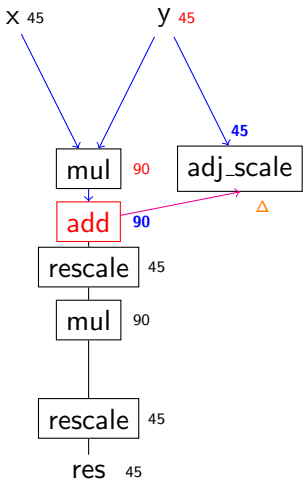
Adjust scale Op

- Example: $(xy + y)^2$
- Addition: scale mismatch
  - xy: scale 90
  - y: scale 45
- Solution: Insert `mgmt.adjust_scale`
  - With *unknown* delta scale Δ



Scale Analysis

- Example: $(xy + y)^2$
- Addition: scale mismatch
  - xy: scale 90
  - y: scale 45
- Solution: Insert `mgmt.adjust_scale`
  - With *unknown* delta scale Δ
- Use `ScaleAnalysis` to determine Δ
  - Forward
    - Add Scale: **90**
    - y Scale: **45**
  - Backward
    - $\Delta = 90 - 45$



Lowering of adjust scale

- Example: $(xy + y)^2$
- Addition: scale mismatch
  - xy: scale 90
  - y: scale 45
- Solution: Insert `mgmt.adjust_scale`
  - With delta scale $\Delta = 45$
- Lower to `mul(y, 1)`
  - 1 is scaled by $\Delta = 45$
  - Does not change message, change scale