Problem: Calls to virtual functions break the control flow of programs and hinder reverse engineering (RE) efforts.

Goal: Statically determine most likely targets of each virtual function call.

Solution: Determine types of objects used in virtual function calls based on how they are used.

1st stage – Binary Analysis and Tracelet Extraction

- Many possible targets
  - **Input**: Many possible targets
  - **Output**: Focus on most likely targets

2nd stage – Train a model for each type

- Use objects for which a type can be statically determined (e.g., recently initialized)
  - **Input**: Many possible targets
  - **Output**: Focus on most likely targets

3rd stage – Compute ranking of types for objects

- Use objects for which a type can’t be statically determined
  - **Input**: Many possible targets
  - **Output**: Focus on most likely targets

Evaluation:
- Evaluated over 20 benchmarks
- Compared to ground truth from manual RE
- Objective: rank expected target highest
- Across all benchmarks, for over 80% of calls to virtual functions, expected target ranked in top 3

Result of *Smoothing.exe*:
- **X** axis – maximum rank
- **Y** axis – percentage of calls to virtual functions for which the expected target was ranked below the maximum rank

The research leading to these results has received funding from the European Union’s - Seventh Framework Programme (FP7) under grant agreement n° 615688 – ERC-COG-PRIME.