Binary Differencing
Dr Andy King, University of Kent, Canterbury
Industrial Fellow, 2008 – 2012

Why do we need to compare binary executables?
• Classify malicious software (malware)
• Discover security vulnerabilities
• Check for infringement against copyright agreements

BinDiff [Halvar Flake, 2005] in a nutshell

function binDiff(G_A, G_B):
S_A ← G_A;
S_B ← G_B;
M' ← Φ;
(M, S_A, S_B) ← initialMatches(S_A, S_B);
while M' ≠ M do
  M ← M;
  (M, S_A, S_B) ← propagateMatches(M, S_A, S_B);
end
return (M, S_A, S_B);

BinDiff example: its solution and its problem

• initialMatches will match vertices 1, 3, 4, 10 and 16: 7 is matched since (7,9,1) is unique, while 10 is unmatched since (6,6,2) is not unique.
• propagateMatches will match vertices 1, 3, 4, 10 and 16 are matched: vertex 4, which is not unique in the graph, but is unique in the subset of children of vertex 9, being its only child
• propagateMatches will match 11 and 12 similarly
• Fails to match 3 vertices: 2, 5, and 15
• Does not quantify degree of structural similarity

BinSlayer [Bourquin, King, Robbins, 2013] in a nutshell

• Compute a bijective function
• Solve an assignment problem \( \phi : V_{G_A} \rightarrow V_{G_B} \)

BinSlayer\( |V_{G_A}| + |V_{G_B}| + |E_{G_A}| + |E_{G_B}| \) und

My Industry Fellowship has completely redefined the direction of my research. Under secondment, I have developed techniques for reasoning about binary programs, a topic that has been largely overlooked by academia. For more details, see the following paper which summaries the body of work that I have undertaken on secondment: Jörg Brauer and Andy King: Transfer Function Synthesis without Quantifier Elimination, Logical Methods in Computer Science, 8(3) August 2012.