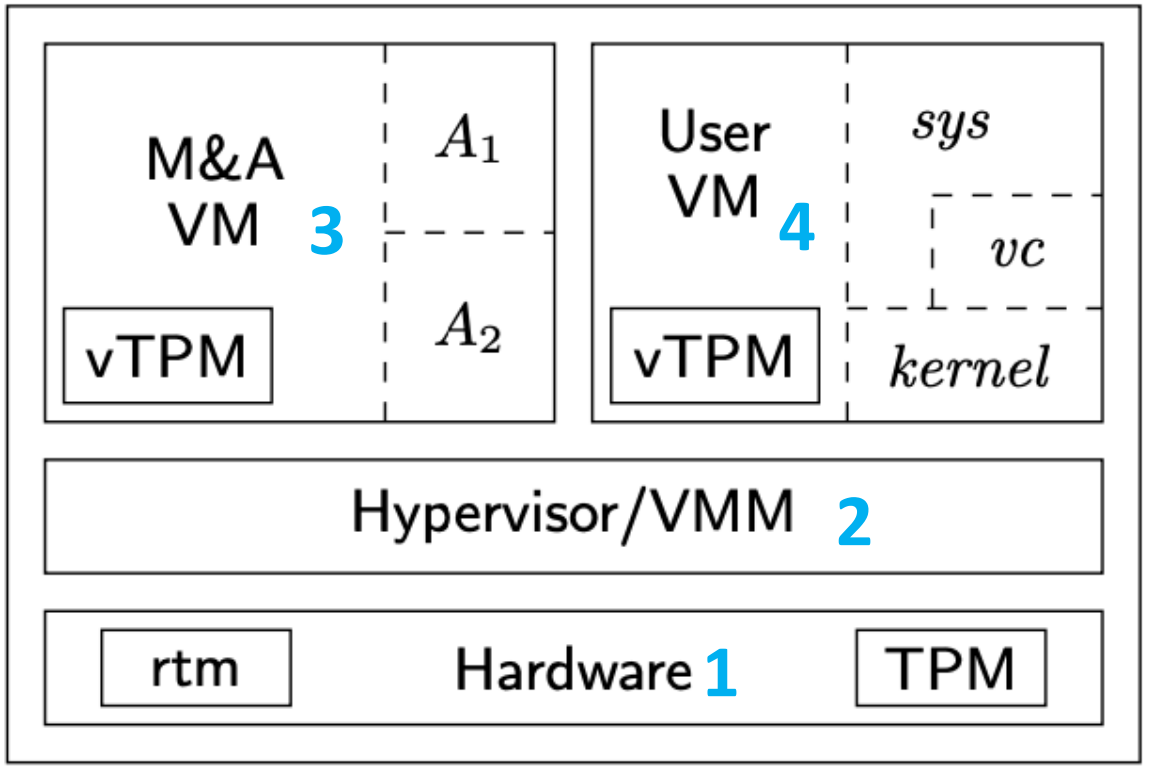
Goals of cost analysis

- Ultimate goal: guide selection of a protocol

- How:
  - systematic variation of assumption
  - assigning cost to each component that's corrupted
    - Assign low (or high?) values to difficult actions
    - Realize set of protocols, one with minimum (maximum) cost
    - Cost may reflect ordering

Say we have the architecture from “Confining the Adversary” Paper

- ms(rtm, A1)
- ms(rtm, A2)
- ms(A1, vc)
- ms(A2, ker)
- msker (vc, sys)



First protocol…. Just measure *sys* using *vc*

- Protocol:
  - @4 [vc 4 sys1]
- Cost?
  - Potentially 2?
    - 2 places where corruptions could occur

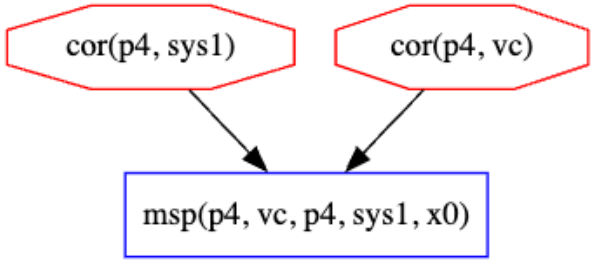
msp(p4, vc, p4, sys1, x0)

Problem Configuration

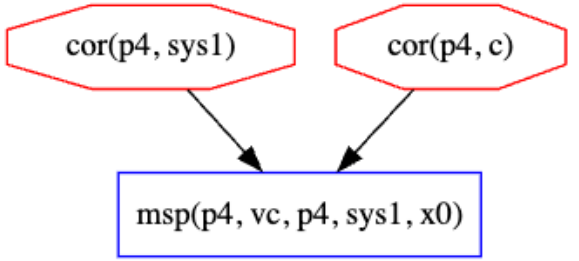
```
[ bound = 500, limit = 5000, input_order ]

% Assume adversary avoids detection at our main measurement
% event. This is a measurement of sys
l(V) = msp(p4, M, p4, sys1, X)
 => corrupt_at(p4, sys1, V).

% Assume no dependencies
depends(p4, C, p4, sys1) => false.

% No recent assumptions

% No deep assumptions

m4_include(`sys.gli')m4_dnl

m4_include(`sys_dist.gli')m4_dnl

m4_include(`thy.gli')m4_dnl
```

Models

Model 1



Model 2



Measure *vc* and *sys* in parallel

- Protocol
  - *target: @p3 [a p4 vc]
    +~+ @p4 [vc p4 sys]
- Cost?
  - Two corruption events and a repair event…
  - What should be the cost of a repair?

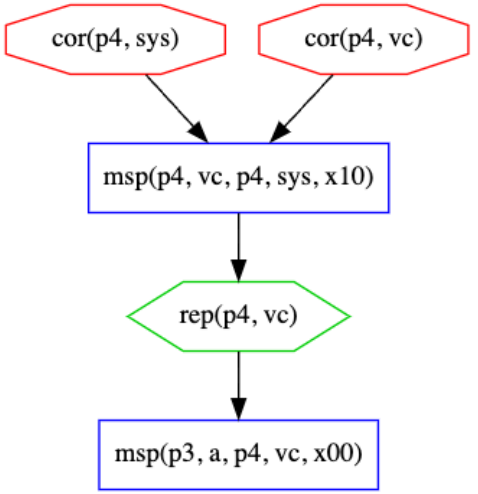
msp(p4, vc, p4, sys, x10) msp(p3, a, p4, vc, x00)

Problem Configuration

```
[ bound = 500, limit = 5000, input_order ]

% Assume adversary avoids detection at
% our main measurement event.
% This is a measurement of sys.
l(V) = msp(p4, M, p4, sys, X)
 => corrupt_at(p4, sys, V).

% Assume sys depends on kernel
depends(p4, C, p4, sys) => false.
depends(p4, C, p4, vc) => false.
depends(p3, C, p3, a) => false.

% Assume no recent corruptions
prec(V, V1) & l(V1) = cor(P,C) & ms_evt(V)
=> false.

% Assume no deep corruptions
l(V) = cor(p3, M) => false.

m4_include(`vc-sys.gli')m4_dnl

m4_include(`vc-sys_dist.gli')m4_dnl

m4_include(`thy.gli')m4_dnl
```

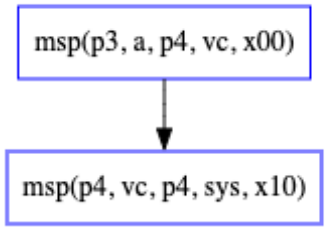
Models

Model 1



Measure *vc* and *sys* in sequence

- Protocol
  - *target: @p3 [a p4 vc]
    +<+ @p4 [vc p4 sys]
- Analysis
  - No models if recent or deep assumption… this is expected

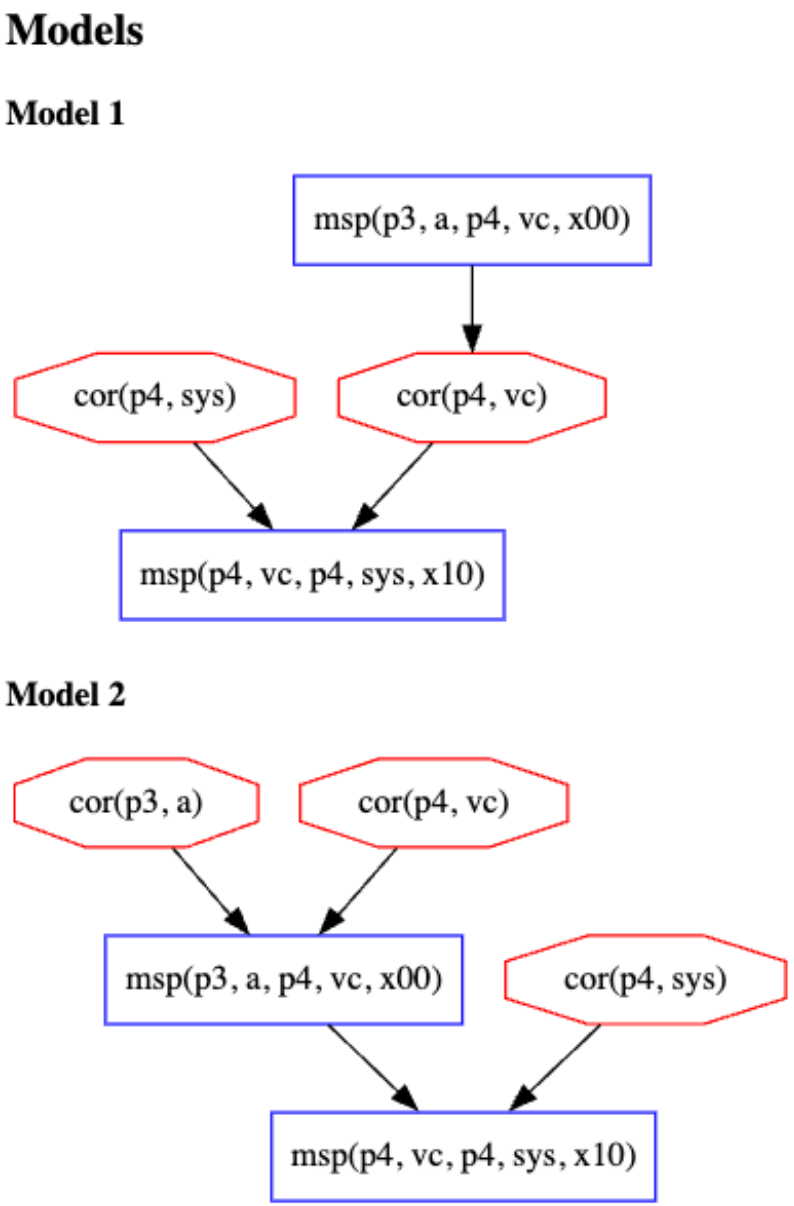


Problem Configuration

```
[ bound = 500, limit = 5000, input_order ]

% Assume adversary avoids detection at
% our main measurement event.
% This is a measurement of sys.
l(V) = msp(p4, M, p4, sys, X)
 => corrupt_at(p4, sys, V).

% Assume sys depends on kernel
% depends(p3, C, p3, a) => C = p1.
depends(p4, C, p4, sys) => false.
depends(p4, C, p4, vc) => false.
depends(p3, C, p3, a) => false.

% Assume no recent corruptions
prec(V, V1) & l(V1) = cor(P,C) & ms_evt(V)
  => false.

% Assume no deep corruptions
l(V) = cor(p3, M) => false.

m4_include(`vc-sys-seq.gli')m4_dnl

m4_include(`vc-sys-seq_dist.gli')m4_dnl

m4_include(`thy.gli')m4_dnl
```

Same protocol…. No recent or deep assumption

- This makes me think… What is the cost of including the recent/deep theorem?

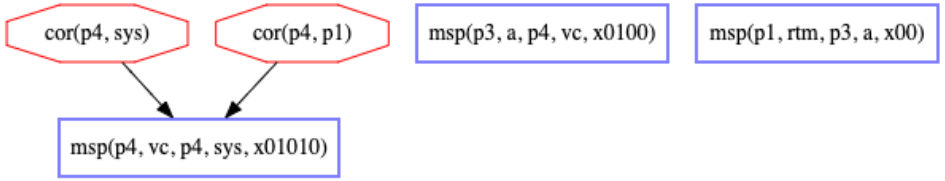


Measure *a* then *vc* then *sys* in parallel

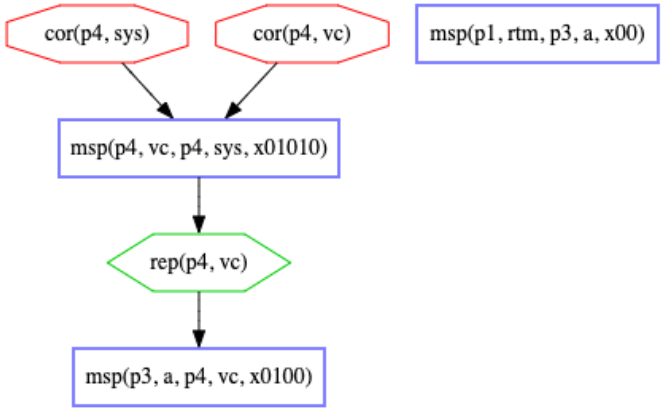
- Protocol
  - *target: @p1 [rtm p3 *a*
    +~+ @p3 [a p4 *vc*
    +~+ @p4 [vc p4 *sys*]]]]

Models

Model 1



Model 2



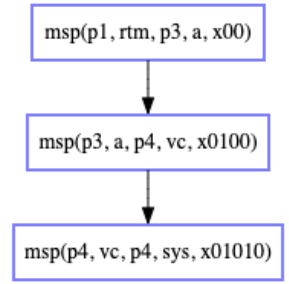
msp(p3, a, p4, vc, x0100) msp(p1, rtm, p3, a, x00) msp(p4, vc, p4, sys, x01010)

Problem Configuration

```
[ bound = 500, limit = 5000, input_order ]

% Assume adversary avoids detection at
% our main measurement event.
% This is a measurement of sys.
l(V) = msp(p4, M, p4, sys, X)
 => corrupt_at(p4, sys, V).

% system dependencies
depends(p3, C, p3, a) => C = p1.
depends(p1, C, p1, rtm) => false.
depends(p4, C, p4, sys) => C = p1.
depends(p4, C, p4, vc) => C = p1.


% Assume no recent corruptions
prec(V, V1) & l(V1) = cor(P,C) & ms_evt(V)
 => false.

% Assume no deep corruptions
l(V) = cor(p3, M) => false.

m4_include(`a-vc-sys-par.gli')m4_dnl

m4_include(`a-vc-sys-par_dist.gli')m4_dnl

m4_include(`thy.gli')m4_dnl
```

Measure *a* then *vc* then *sys* in sequence

- Protocol
  - *target: @p1 [rtm p3 *a*
    +<+ @p3 [a p4 *vc*
    +<+ @p4 [vc p4 *sys*]]]]

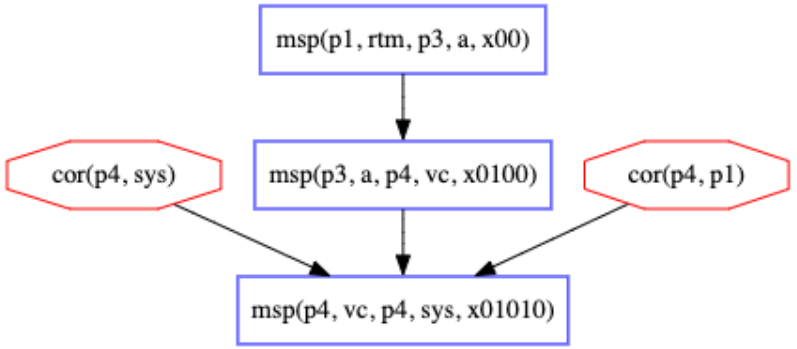


Problem Configuration

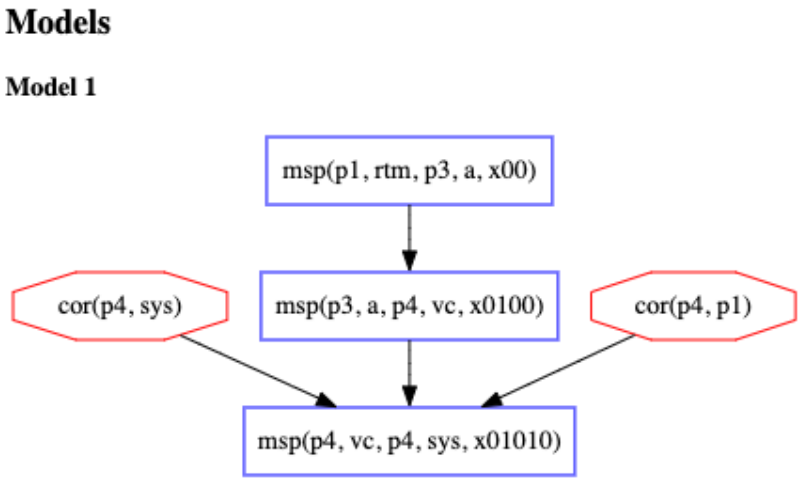
```
[ bound = 500, limit = 5000, input_order ]

% Assume adversary avoids detection at
% our main measurement event.
% This is a measurement of sys.
l(V) = msp(p4, M, p4, sys, X)
 => corrupt_at(p4, sys, V).

% Assume sys depends on kernel
depends(p3, C, p3, a) => C = p1.
depends(p1, C, p1, rtm) => false.
depends(p4, C, p4, sys) => C = p1.
depends(p4, C, p4, vc) => C = p1.

% Assume no recent corruptions
prec(V, V1) & l(V1) = cor(P,C) & ms_evt(V)
  => false.

% Assume no deep corruptions
l(V) = cor(p3, M) => false.

m4_include(`a-vc-sys-seq.gli')m4_dnl

m4_include(`a-vc-sys-seq_dist.gli')m4_dnl

m4_include(`thy.gli')m4_dnl
```

Models

Model 1



Thoughts/Takeaways



- Cost of adding an assumption?
- Cost of adding a dependencies?
- Cost of applying recent/deep theorem?
  - Should we consider this a standard assumption?

- Cost of a corruption/repair event?
  - Maybe turn protocol execution into a tree… then could look at depth of corruption/repair event and that could be the event's cost. Sum all costs together and that is the total cost.