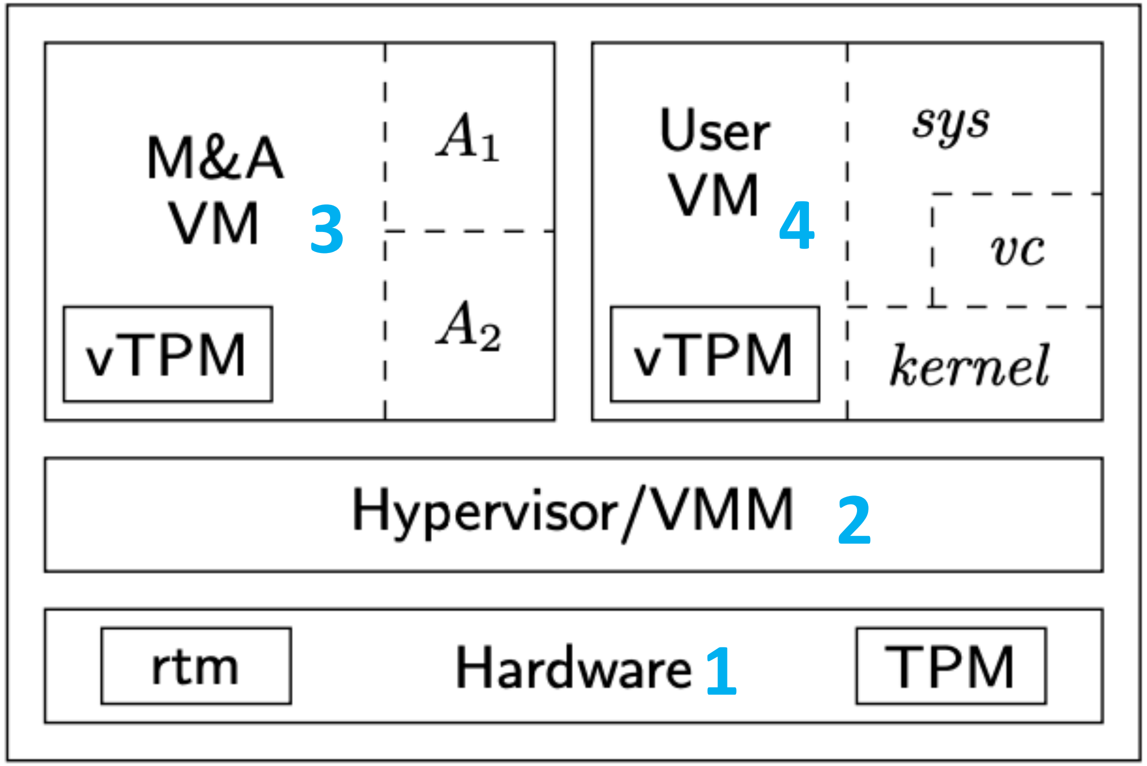
Goals of cost analysis

- 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

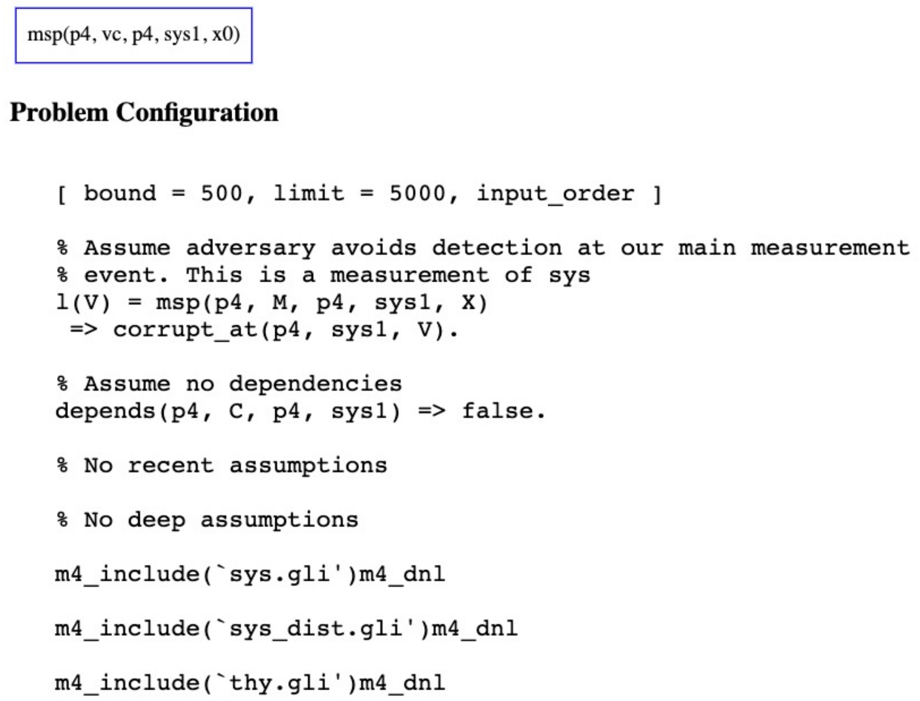
- ms(rtm, A1)
- ms(rtm, A2)
- ms(A1, vc)
- ms(A2, ker)
- msker (vc, sys)



Control Variables

- Assumptions
    - Always assume recent/deep
    - Make no assumptions about system dependencies (except maybe that the TPM is the root of trust)

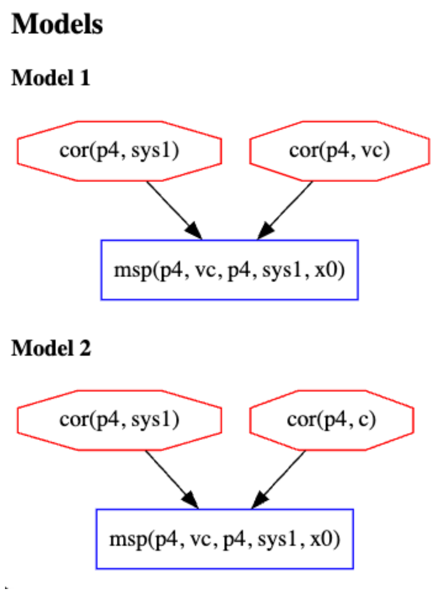
First protocol…. Just measure *sys* using *vc*



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

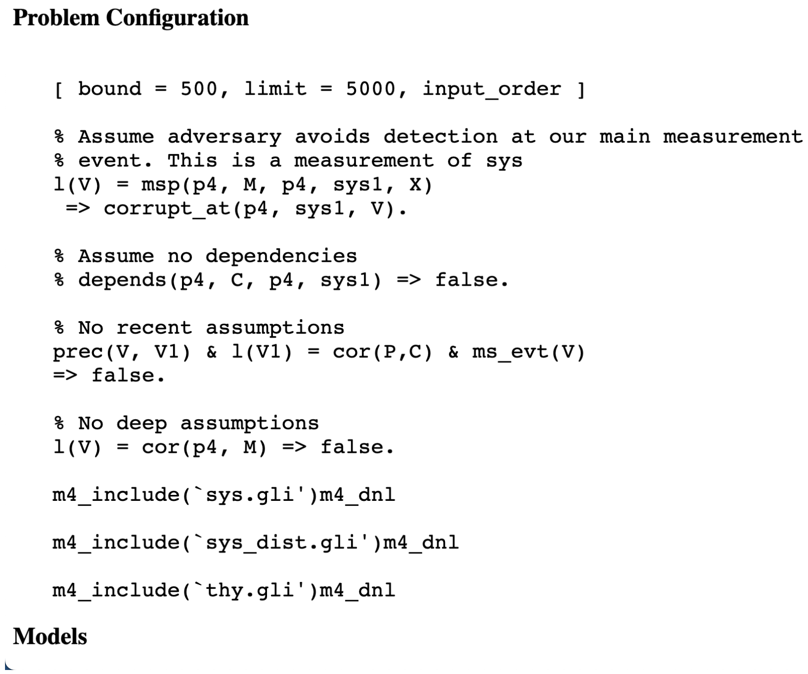
Model 1

Model 2

Event	Cost
cor(p4,sys1)	c1
cor(p4,vc)	c2
TOTAL COST	c1+c2

First protocol…. With recent or deep assumptions

- No models… no cost.



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
prec(V, V1) & l(V1) = cor(P,C) & ms_evt(V)
=> false.

% No deep assumptions
l(V) = cor(p4, M) => false.

m4_include(`sys.gli')m4_dnl

m4_include(`sys_dist.gli')m4_dnl

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

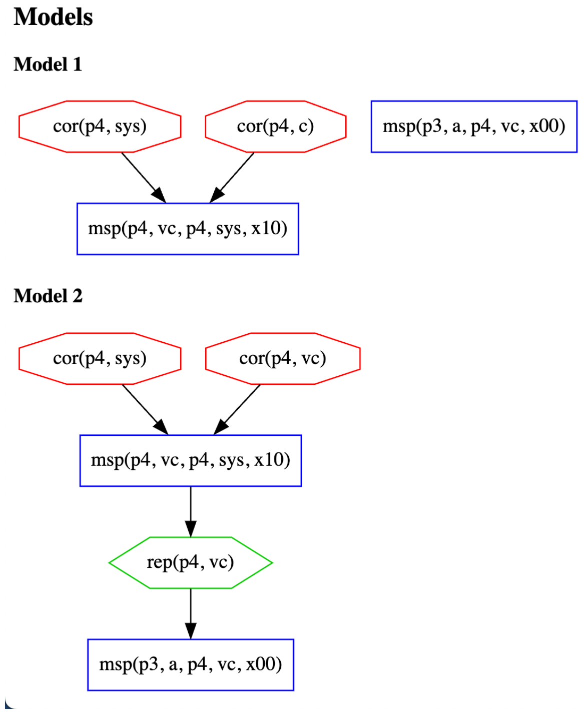
Models

Event	Cost
TOTAL COST	0

Measure *vc* and *sys* in parallel

- Protocol
  - *target: @p3 [a p4 vc]

    +~+ @p4 [vc p4 sys]



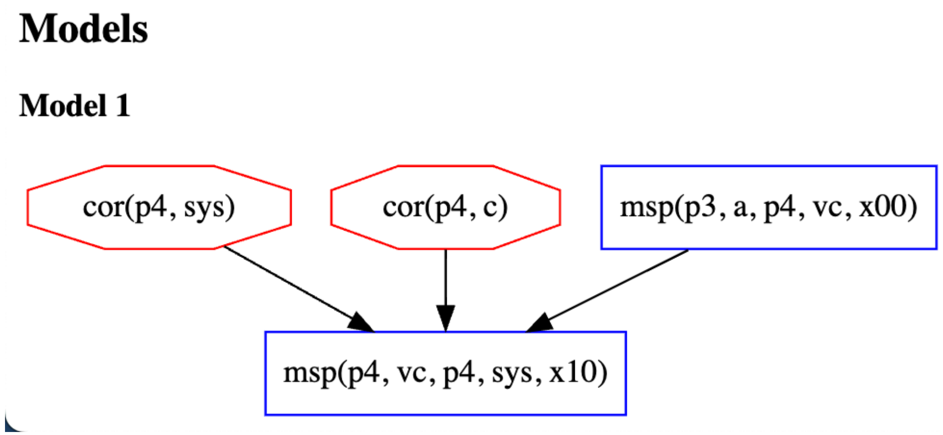
Model 1	
Event	Cost
cor(p4,sys)	c1
cor(p4,c)	c4
TOTAL COST	c1+c4

Model 2	
Event	Cost
cor(p4,sys)	c1
cor(p4,vc)	c2
rep(p4,vc)	c3
TOTAL COST	c1+c2+c3

Measure *vc* and *sys* in sequence

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

Model 1	
Event	Cost
cor(p4,sys)	c1
cor(p4,c)	c4
TOTAL COST	c1+c4



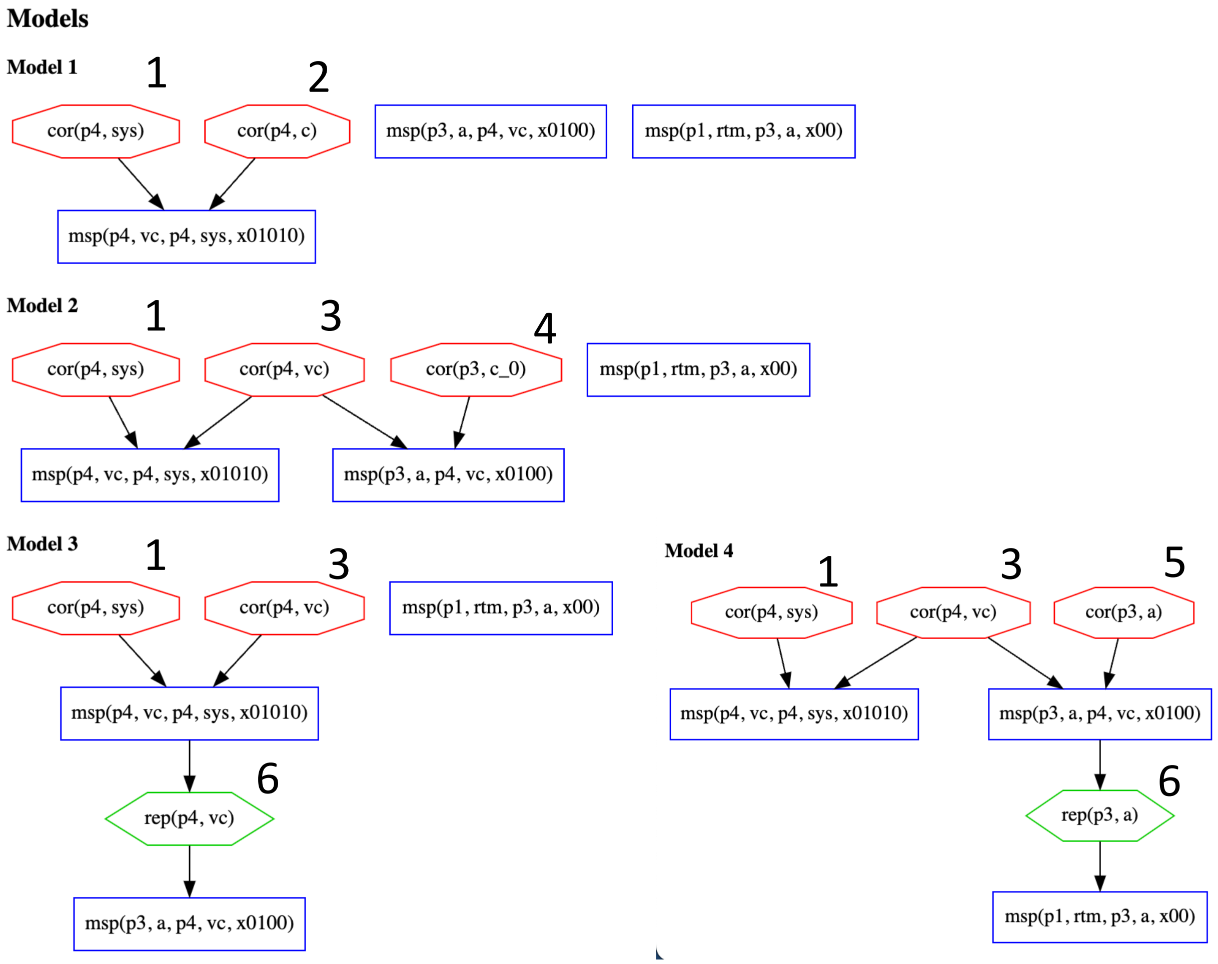
Models

Model 1

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

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

Model	Total cost
1	c1 + c2
2	c1 + c2 + c4
3	c1 + c3 + r6
4	c1 + c3 + c5 + r6



Models

Model 1

1 cor(p4, sys)

2 cor(p4, c)

msp(p3, a, p4, vc, x0100)

msp(p1, rtm, p3, a, x00)

msp(p4, vc, p4, sys, x01010)

Model 2

1 cor(p4, sys)

3 cor(p4, vc)

4 cor(p3, c_0)

msp(p1, rtm, p3, a, x00)

msp(p4, vc, p4, sys, x01010)

msp(p3, a, p4, vc, x0100)

Model 3

1 cor(p4, sys)

3 cor(p4, vc)

msp(p1, rtm, p3, a, x00)

msp(p4, vc, p4, sys, x01010)

6 rep(p4, vc)

msp(p3, a, p4, vc, x0100)

Model 4

1 cor(p4, sys)

3 cor(p4, vc)

5 cor(p3, a)

msp(p4, vc, p4, sys, x01010)

msp(p3, a, p4, vc, x0100)

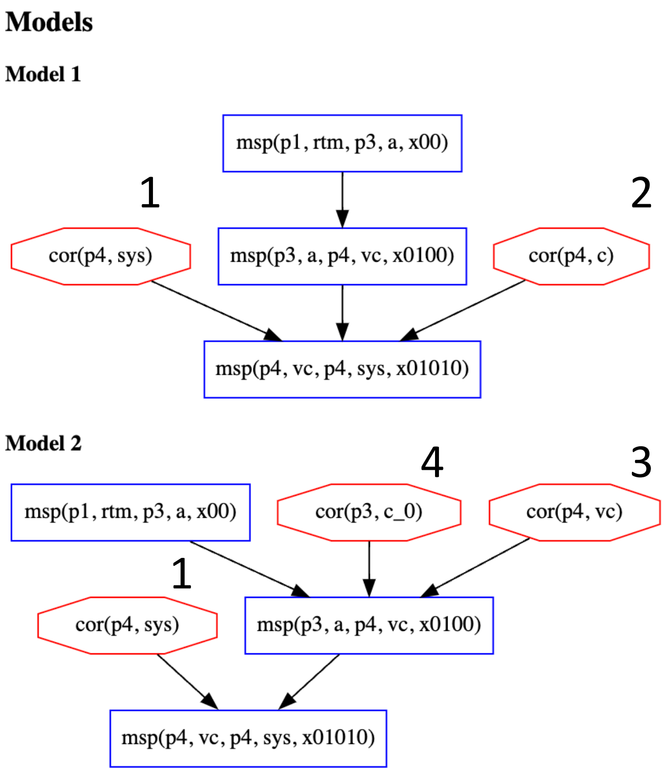
6 rep(p3, a)

msp(p1, rtm, p3, a, x00)

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

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

Model	Total cost
1	c1 + c2
2	c1 + c3 + c4



Thoughts/Takeaways