Secure Protocols via AI and CPSA

Lauren Brandt

Mentors: Dr. Joshua Guttman and Dr. Andres Molina-Markham

Research Program Lead: Gabby Raymond

HCSS
May 13, 2025

MITRE

Overview

Improving the usability of CPSA.

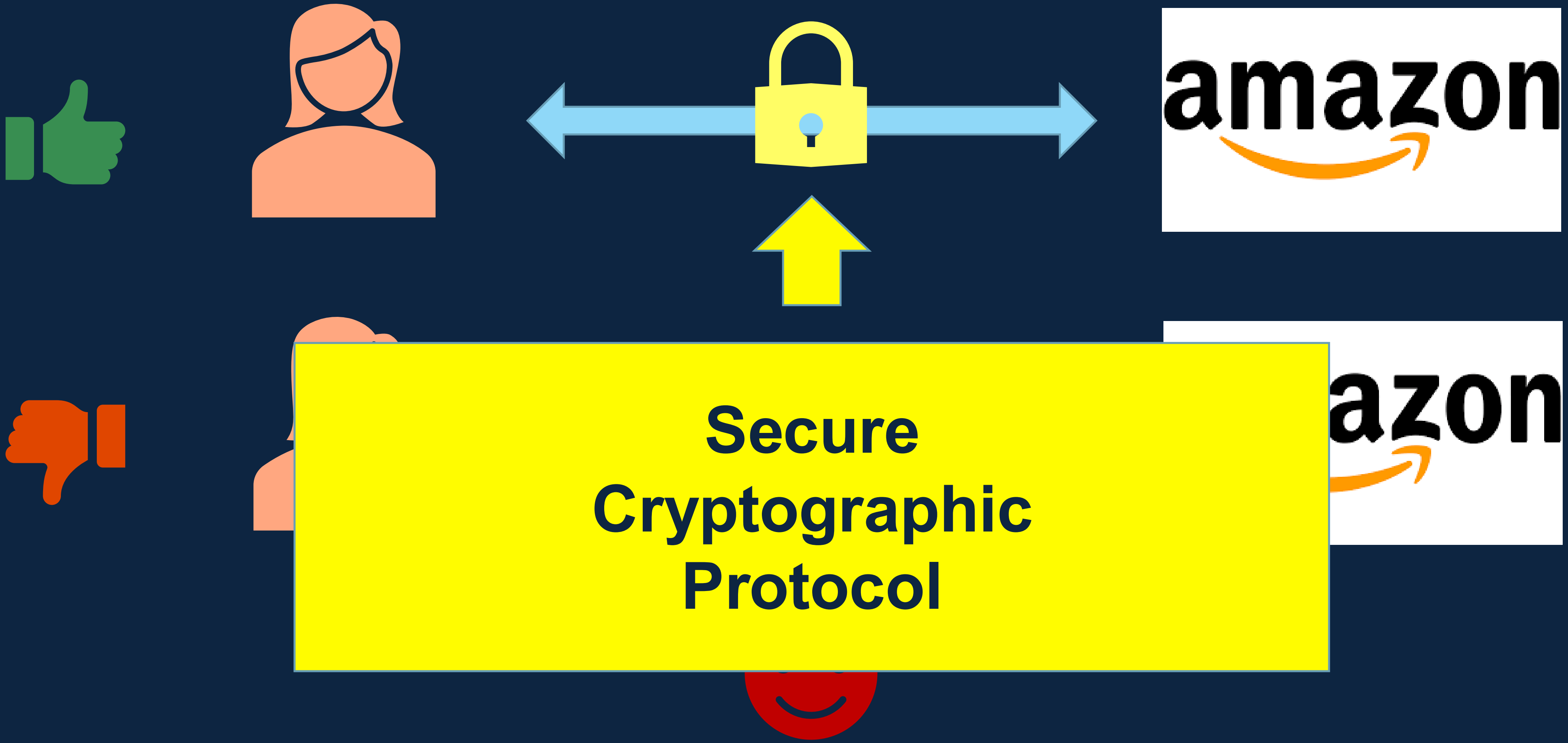
Combining formal methods with LLMs.

- **LLMs for CPSA:** LLMs act as translators and assistants, making CPSA more user-friendly for both experts and novices.
- **CPSA for LLMs:** Risk minimal: CPSA verifies the LLM outputs
- **Method Highlights:** Natural language (NL) format for specifying protocols, reliable and automatic scoring method

LLM – Large Language Model
CPSA – Cryptographic Protocol Shapes Analyzer

MITRE

Cryptographic Protocol



Secure Cryptographic Protocol

Hard to Design Protocols

SSH	IPsec	TLS 1.3	DTLS 1.3
24 Draft Versions	10 Draft Versions	28 Draft Versions	43 Draft Versions
9 Years	3 Years	4 Years	6 Years

Long and **error prone** process to develop these secure protocol standards

Are they using any tools to create these protocols? Yes.

Can LLMs help make these tools more usable? Maybe.

Tamarin

Isabelle

ProVerif

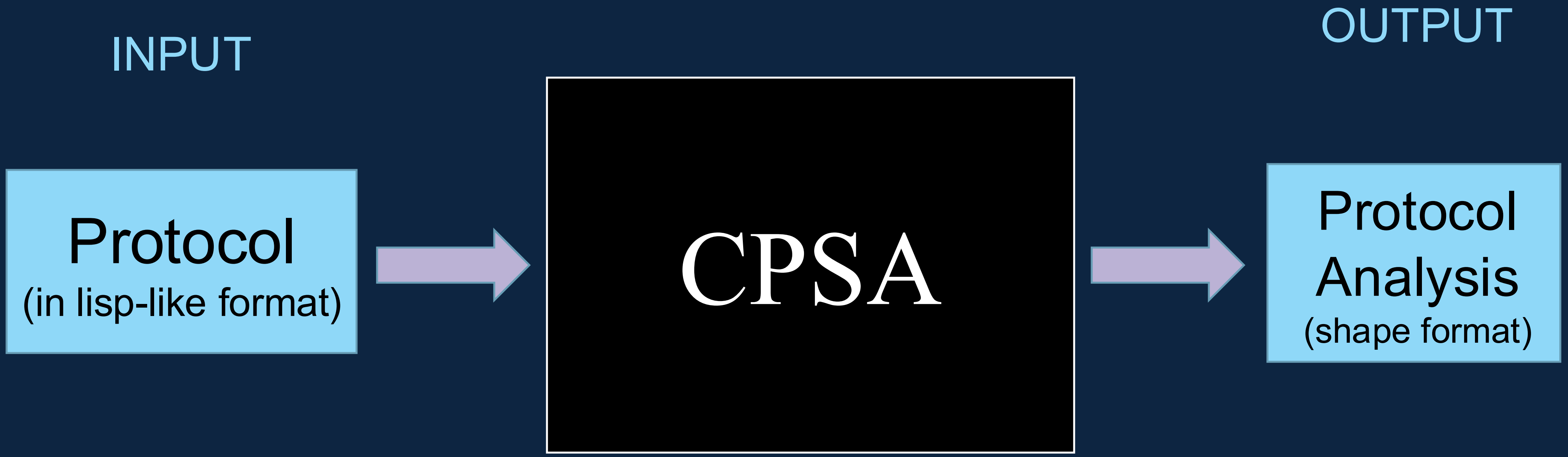
CPSA

But they can be <u>very hard</u> to use.

(They can be so difficult to use, in fact, the international formal methods community has created a working group that aims to make these tools more usable)

MITRE

Cryptographic Protocol Shapes Analyzer (CPSA)

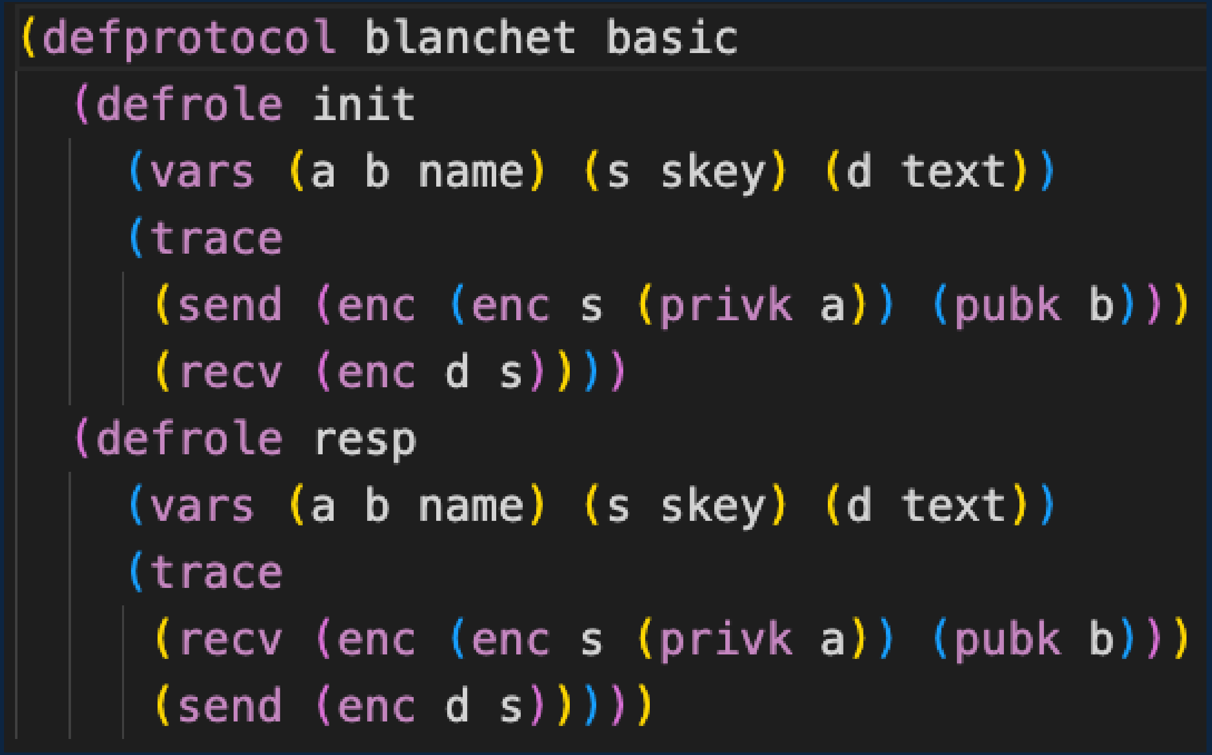


Shapes in the CPSA analysis reveals authentication and confidentiality properties about the protocol.

Shapes = all essentially different runs of a protocol in a strong adversary environment

Cryptographic Protocol Shapes Analyzer (CPSA)

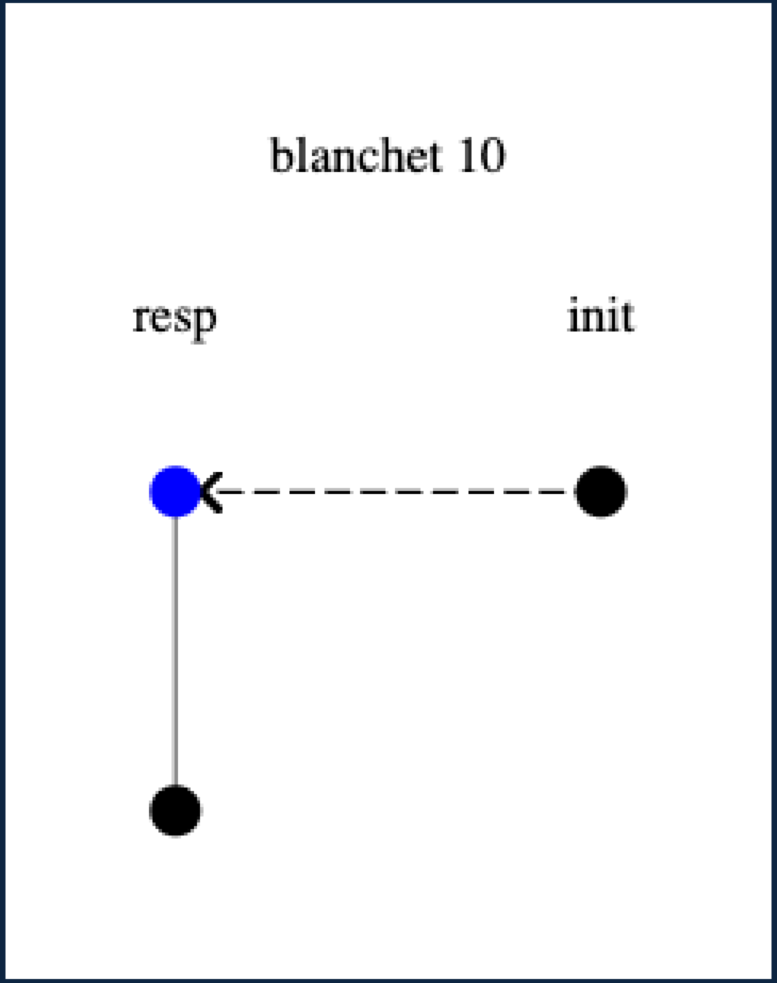
INPUT



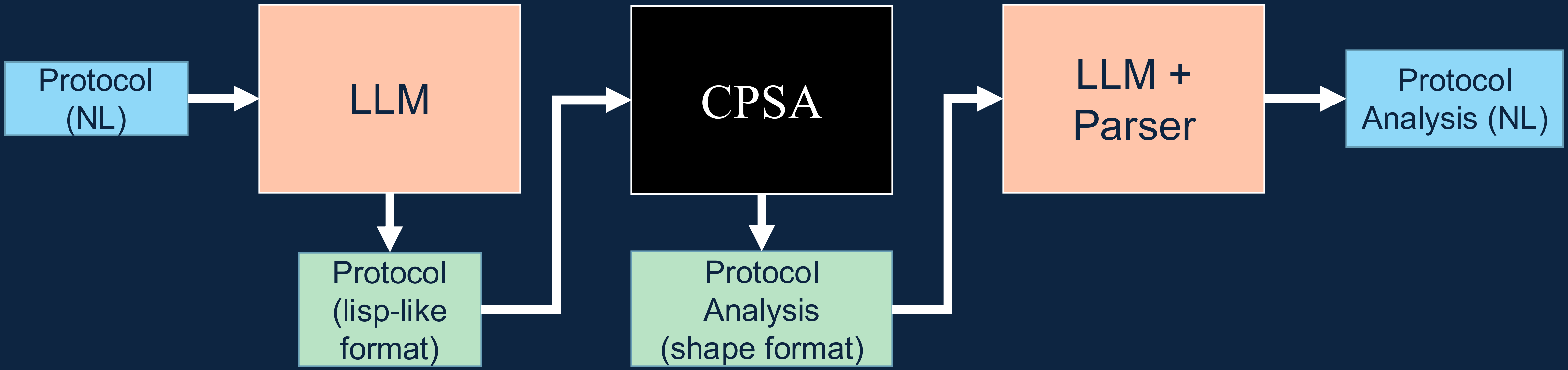
```
(defprotocol blanchet basic
  (defrole init
    (vars (a b name) (s skey) (d text))
    (trace
     (send (enc (enc s (privk a)) (pubk b)))
     (recv (enc d s))))
  (defrole resp
    (vars (a b name) (s skey) (d text))
    (trace
     (recv (enc (enc s (privk a)) (pubk b)))
     (send (enc d s)))))
```

These can be hard
to create and understand
Can LLMs Help?

OUTPUT



Technical Approach

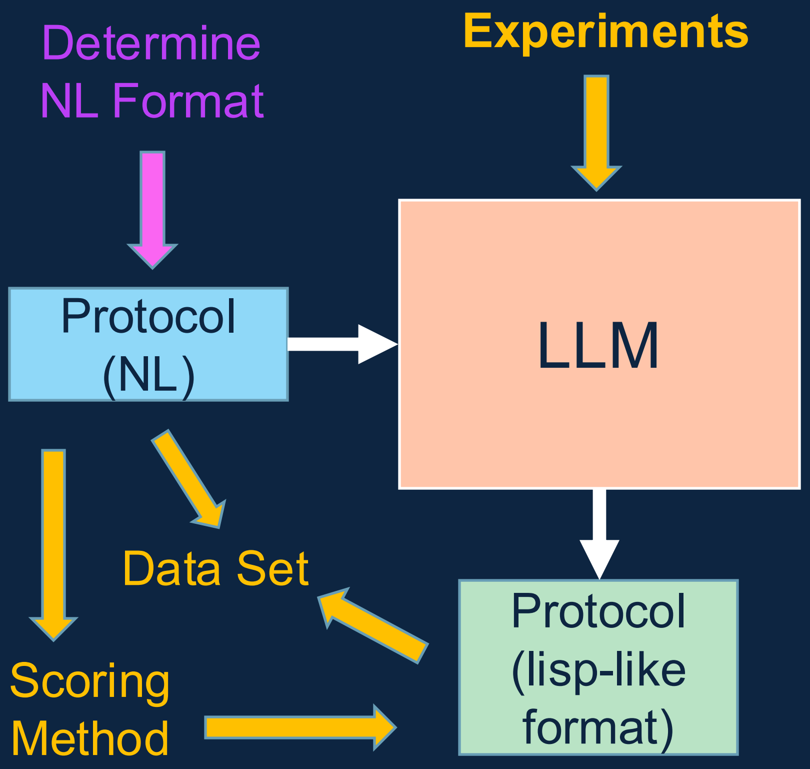


NL – Natural Language
LLM – Large Language Model
CPSA – Cryptographic Protocol Shapes Analyzer

LLMs acting as a translator between natural language and CPSA input/output formats.

MITRE

NL → CPSA LLM



Main Questions:

1. How effective are LLMs at this translation?
2. How does this benefit a CPSA user?
3. How do we handle hallucinations?

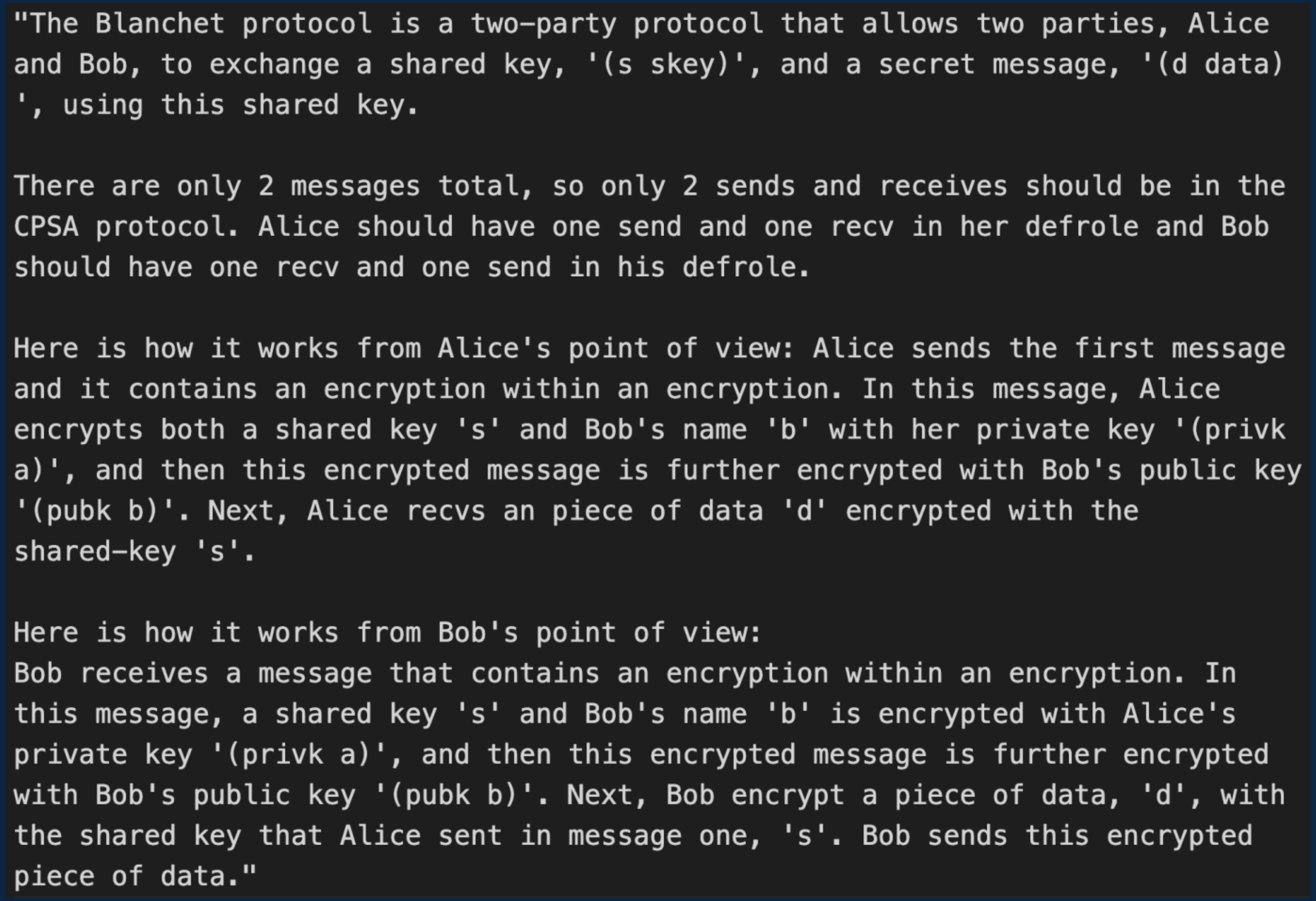
MITRE

The NL Format

Basic overview of protocol

Number of messages

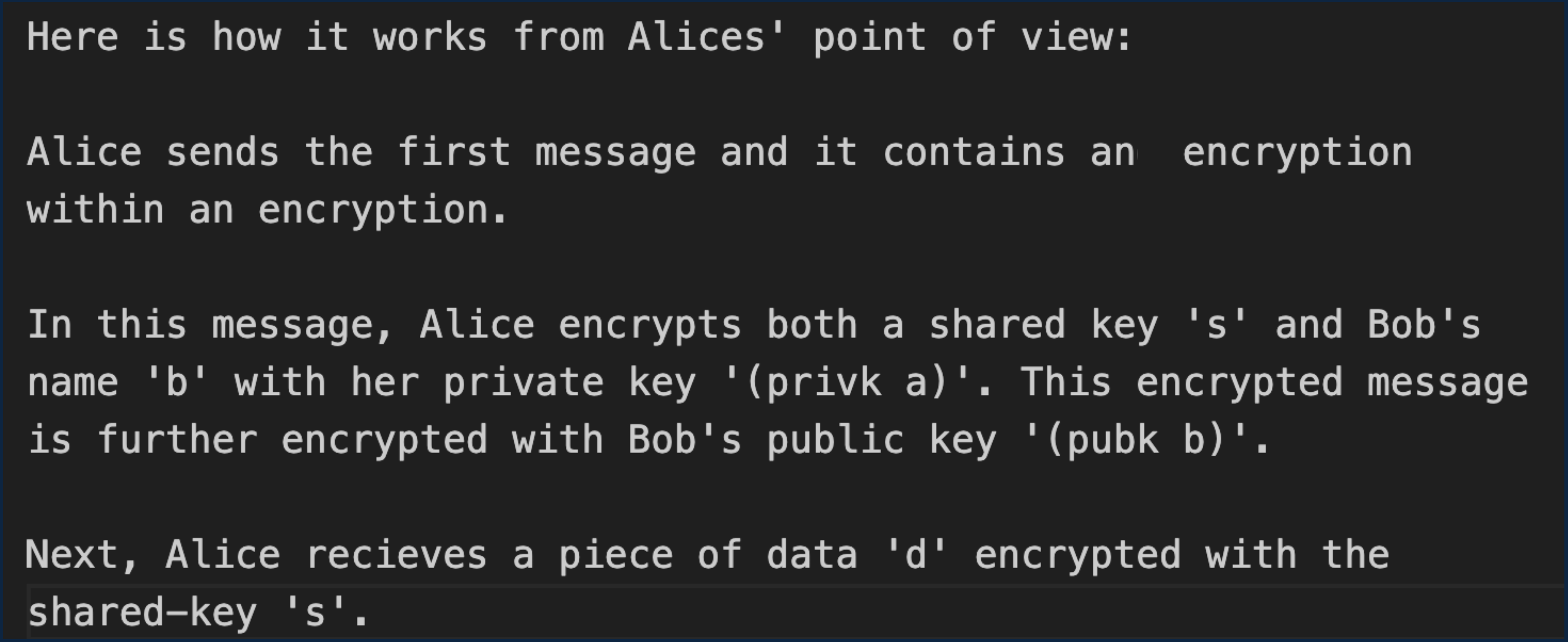
Clear description of protocol from **each role's POV**, (with careful wording for the double encryption.)



```
"The Blanchet protocol is a two-party protocol that allows two parties, Alice
and Bob, to exchange a shared key, '(s skey)', and a secret message, '(d data)
', using this shared key.

There are only 2 messages total, so only 2 sends and receives should be in the
CPSA protocol. Alice should have one send and one recv in her defrole and Bob
should have one recv and one send in his defrole.

Here is how it works from Alice's point of view: Alice sends the first message
and it contains an encryption within an encryption. In this message, Alice
encrypts both a shared key 's' and Bob's name 'b' with her private key '(privk
a)', and then this encrypted message is further encrypted with Bob's public key
'(pubk b)'. Next, Alice recvs an piece of data 'd' encrypted with the
shared-key 's'.

Here is how it works from Bob's point of view:
Bob receives a message that contains an encryption within an encryption. In
this message, a shared key 's' and Bob's name 'b' is encrypted with Alice's
private key '(privk a)', and then this encrypted message is further encrypted
with Bob's public key '(pubk b)'. Next, Bob encrypt a piece of data, 'd', with
the shared key that Alice sent in message one, 's'. Bob sends this encrypted
piece of data."
```

MITRE

The NL Format – Zooming In

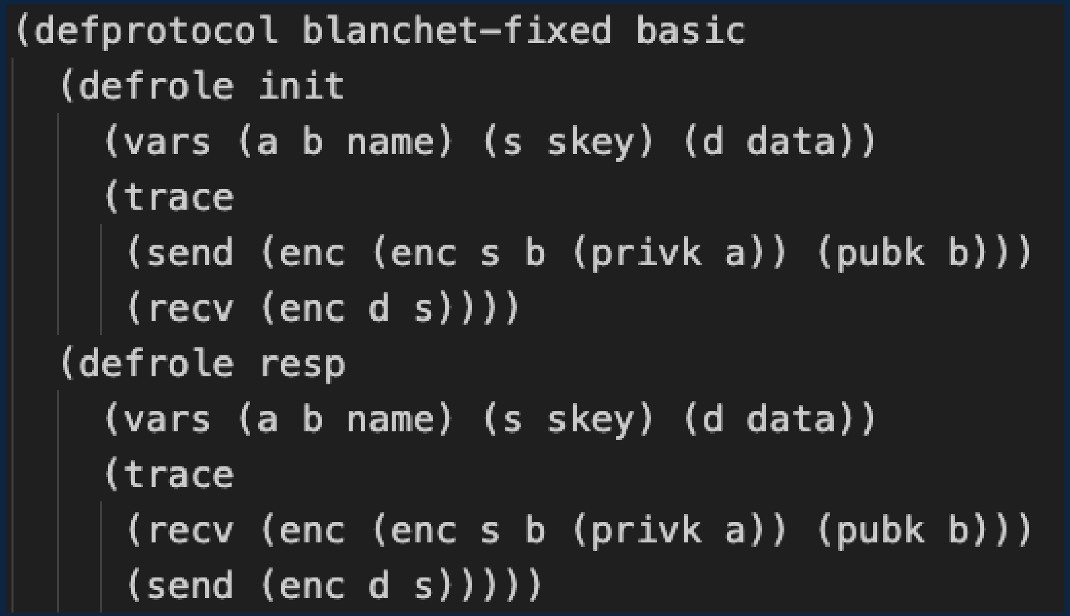


```
Here is how it works from Alices' point of view:

Alice sends the first message and it contains an  encryption
within an encryption.

In this message, Alice encrypts both a shared key 's' and Bob's
name 'b' with her private key '(privk a)'. This encrypted message
is further encrypted with Bob's public key '(pubk b)'.

Next, Alice recieves a piece of data 'd' encrypted with the
shared-key 's'.
```

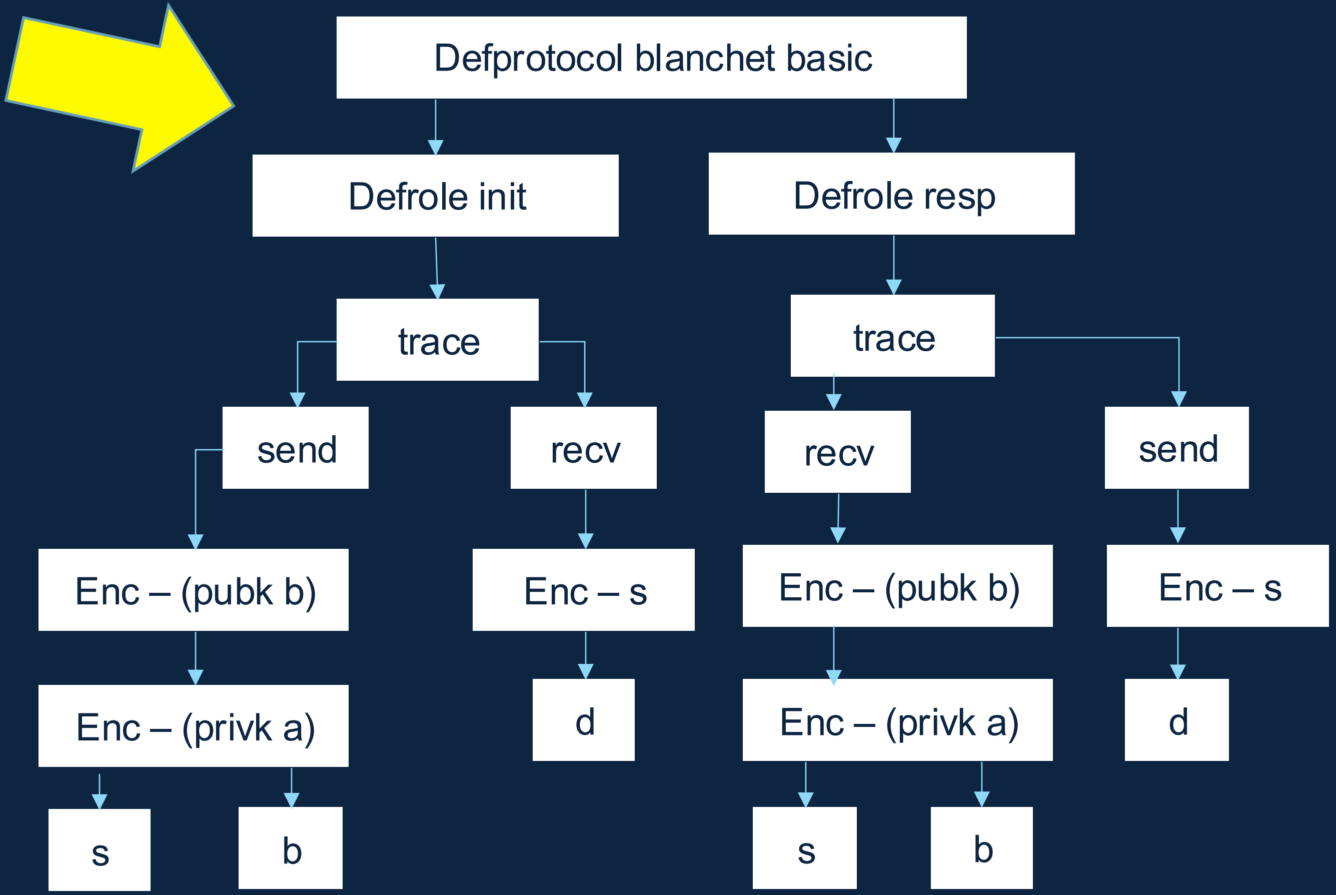
If the user understands the protocol,
the NL format should not be difficult to write.

Scoring Method

- **Goal:** Evaluate how accurately a LLM translates protocol messages.
- First concerned about semantics:
  - LLMs already match the CPSA formatting very closely; CPSA to help with any syntax issues
  - This process is completely unhelpful if the generated protocols run through CPSA without error, but the protocol is completely different from the user's intent.
  - Long-term concerned about both semantics and syntax
- Idea: Decompose CPSA into trees and calculate the tree edit distance between actual and generated CPSA

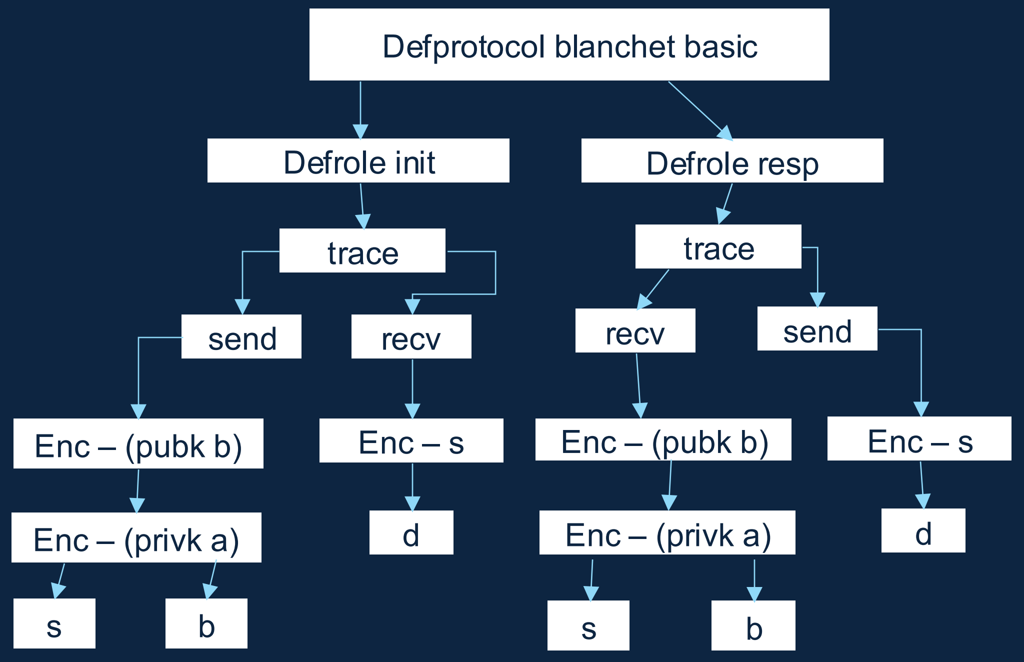


```
(defprotocol blanchet-fixed basic
  (defrole init
    (vars (a b name) (s skey) (d data))
    (trace
     (send (enc (enc s b (privk a)) (pubk b)))
     (recv (enc d s))))
  (defrole resp
    (vars (a b name) (s skey) (d data))
    (trace
     (recv (enc (enc s b (privk a)) (pubk b)))
     (send (enc d s)))))
```



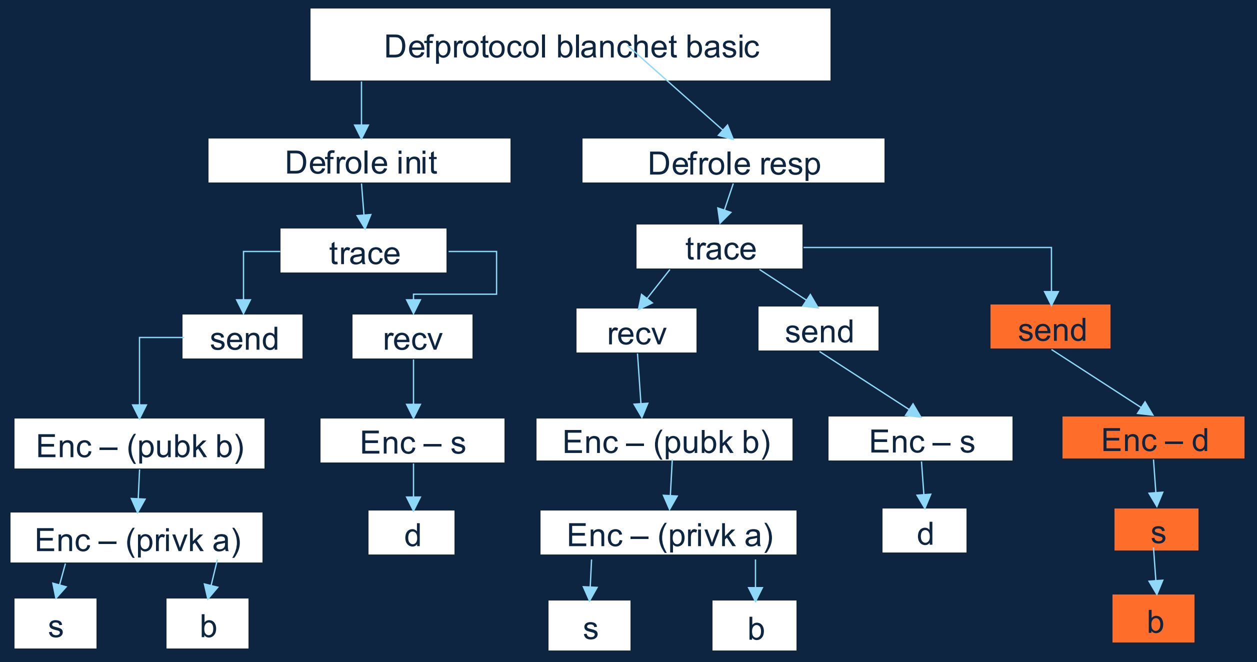
CPSA can be decomposed into trees.

Actual CPSA Tree



Full tree cost: 21

Generated CPSA Tree



Tree edit distance: 4

Find the <u>tree edit distance</u> between actual and generated CPSA trees.

Tree edit distance: Minimum number of additions, deletions, or modifications to turn one tree into another

MITRE

Experiments

GPT-4o
GPT-4o-mini
Codellama-13b-Instruct

Model Experiment

+

LAST

Prompt to LLM:

Here are **two** commented examples of a NL protocol paired with it's CPSA translation.

Use this CPSA context to help translate.

Here is a NL description of a new protocol.
Translate the NL to CPSA.

Prompt Experiment

+

Non-role based
Role-based

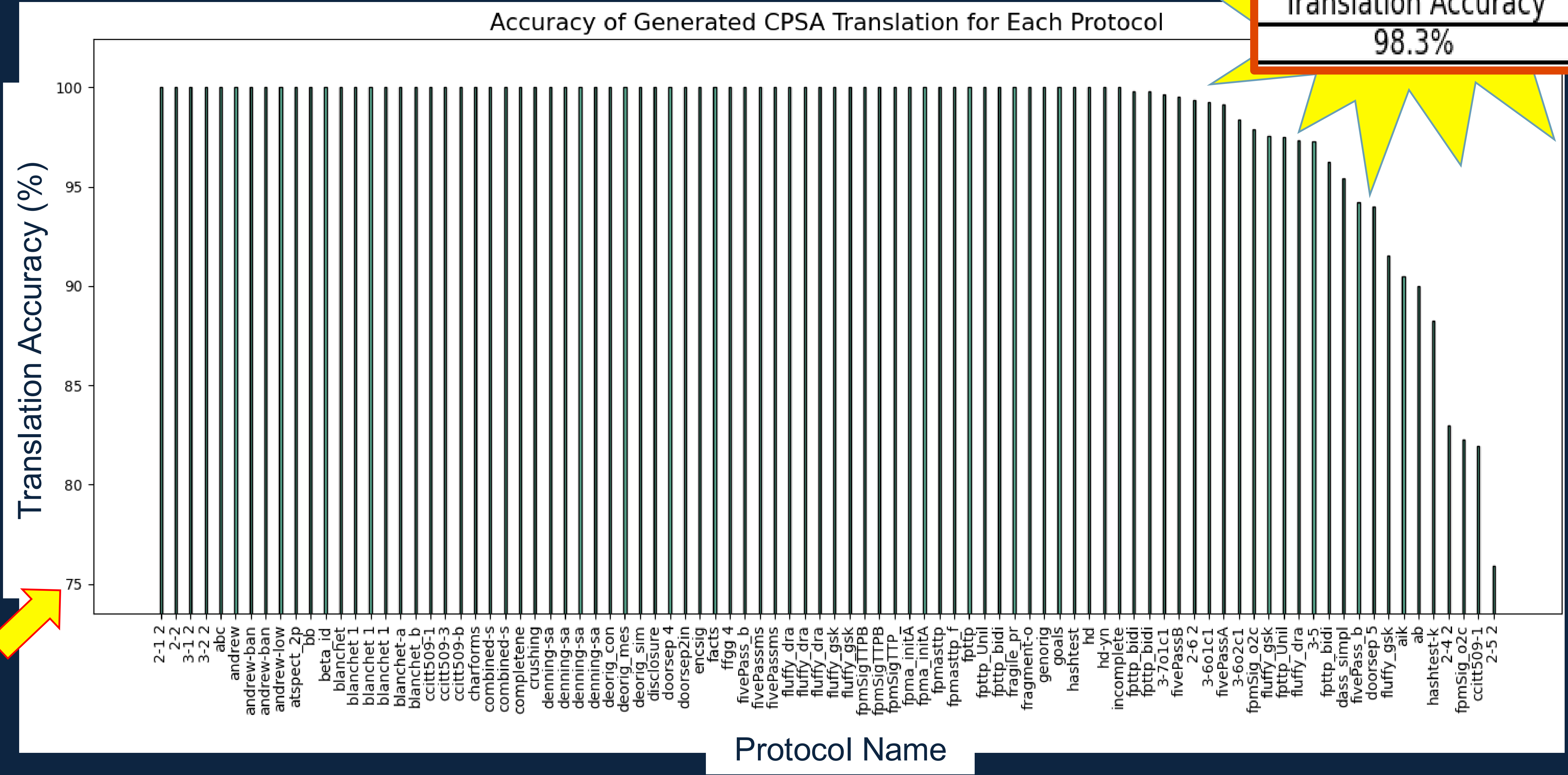
Context Example Format Experiment

+

90 Protocols

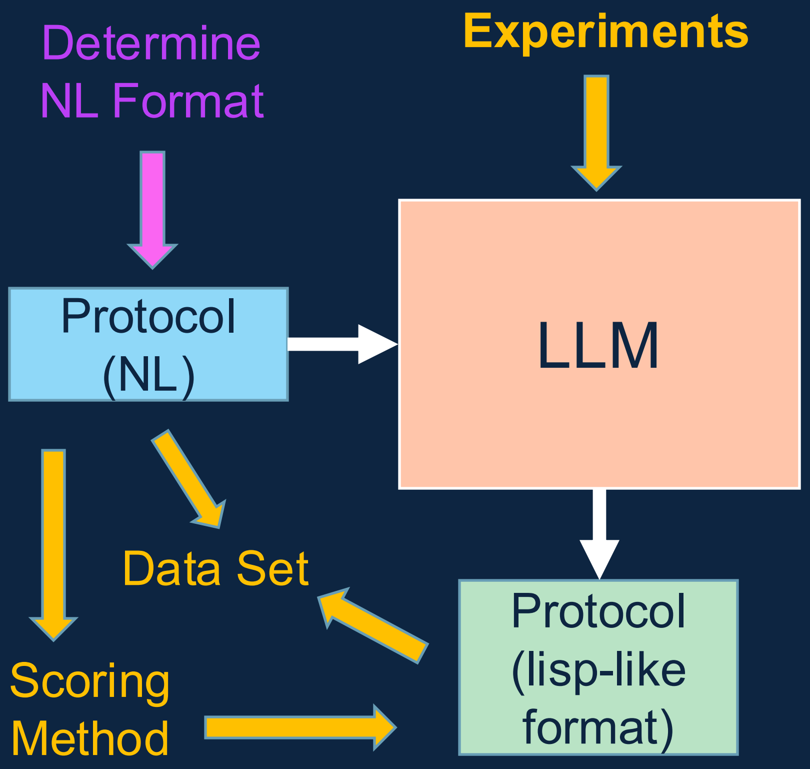
= ? ?

Translation Accuracy
98.3%



MITRE

NL → CPSA Summary



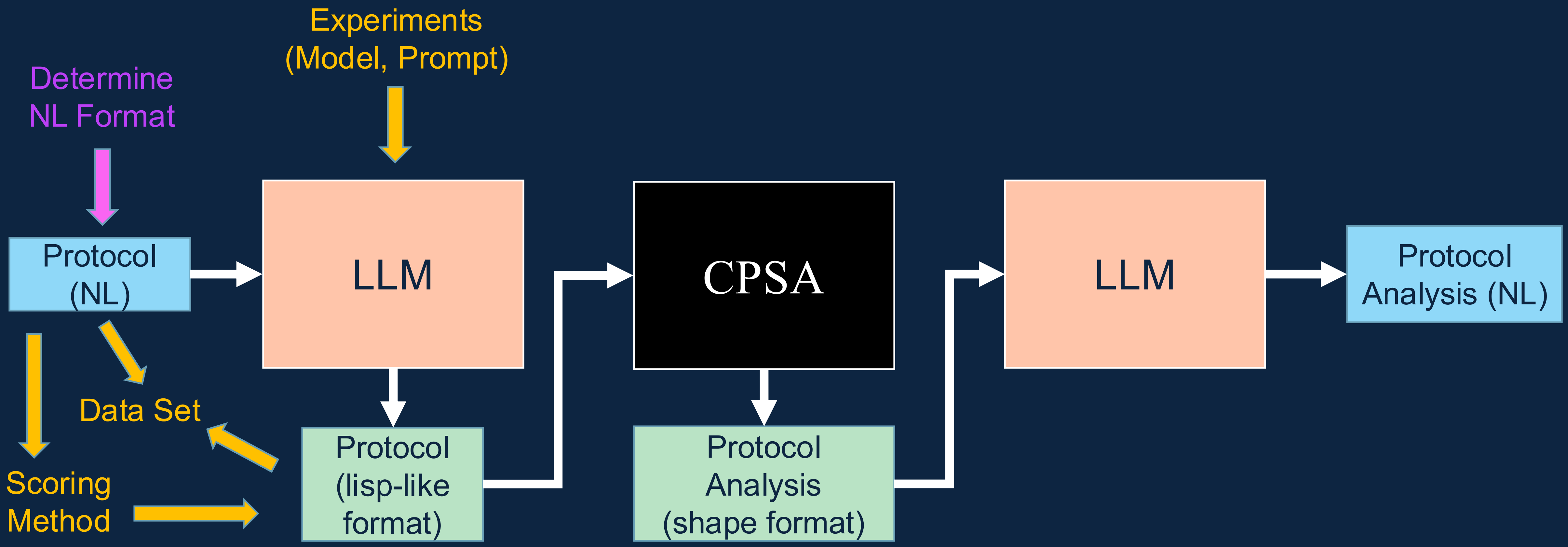
1. How effective are LLMs at this translation?
   - Very effective – 98.3% translation accuracy

2. How does this benefit a CPSA user?
   - For the novice: Easier to learn and use CPSA
   - For the expert: Significantly minimizes manual effort

3. How do we handle hallucinations?
   - Accept the error – new protocol
   - Trace down and fix the error – which we already do

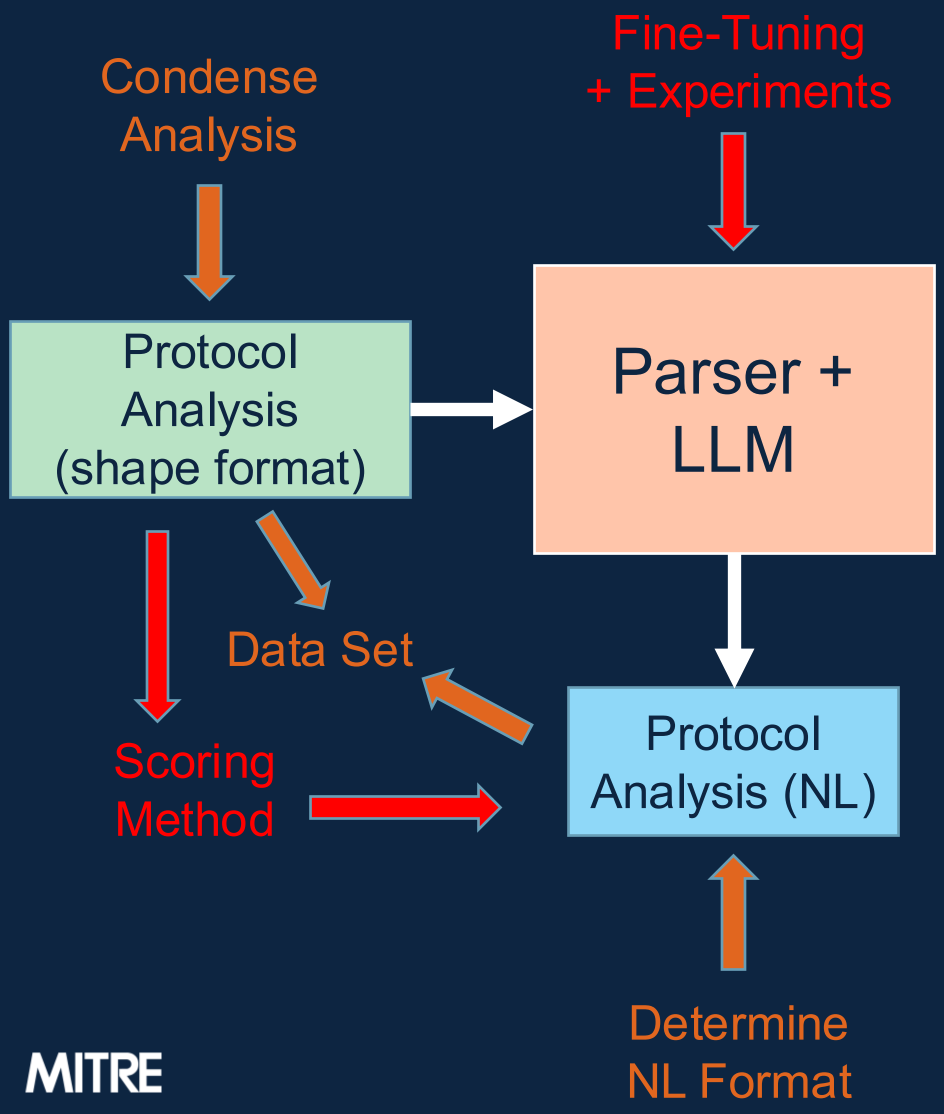
MITRE

Research Plan Progress



MITRE

CPSA → NL Main Question

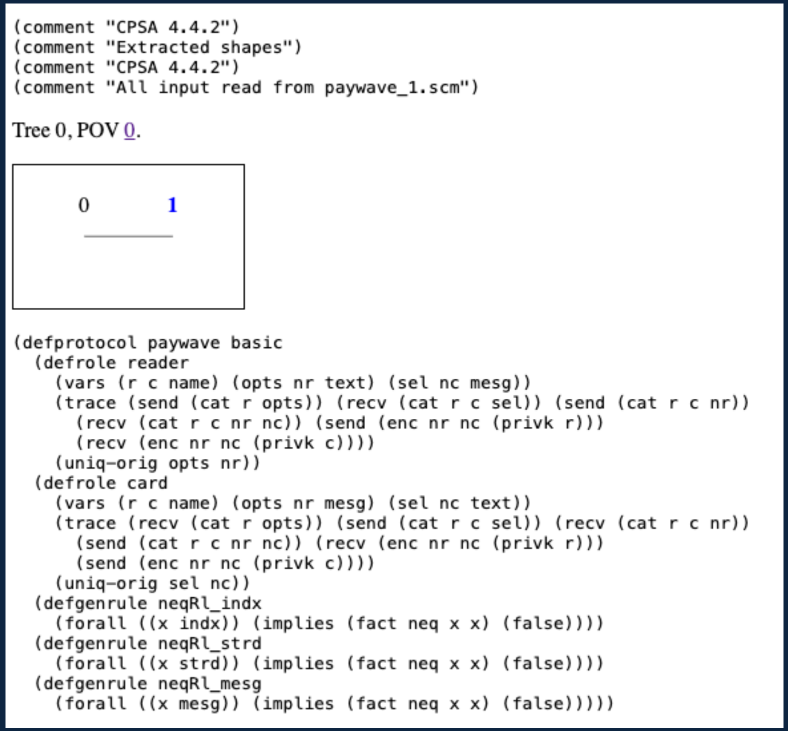


How can we use LLMs and other techniques to:

- identify protocol weaknesses
- strengthen them

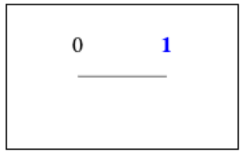
MITRE

Determine NL Format

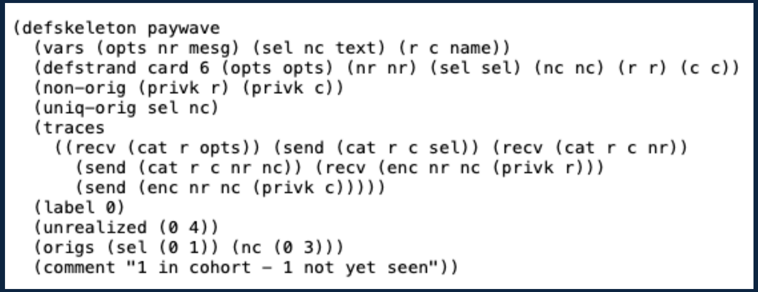


```
(comment "CPSA 4.4.2")
(comment "Extracted shapes")
(comment "CPSA 4.4.2")
(comment "All input read from paywave_1.scm")
```

Tree 0, POV 0.

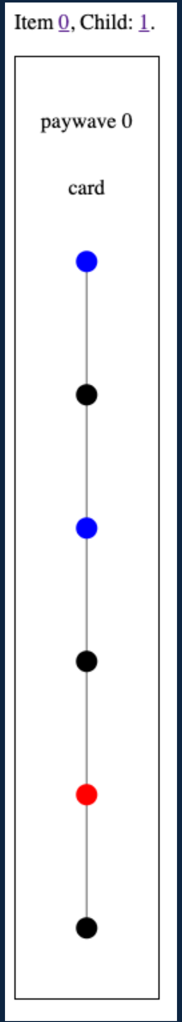


```
(defprotocol paywave basic
  (defrole reader
    (vars (r c name) (opts nr text) (sel nc mesg))
    (trace (send (cat r opts)) (recv (cat r c sel)) (send (cat r c nr))
      (recv (cat r c nr nc)) (send (enc nr nc (privk r)))
      (recv (enc nr nc (privk c))))
    (uniq-orig opts nr))
  (defrole card
    (vars (r c name) (opts nr mesg) (sel nc text))
    (trace (recv (cat r opts)) (send (cat r c sel)) (recv (cat r c nr))
      (send (cat r c nr nc)) (recv (enc nr nc (privk r)))
      (send (enc nr nc (privk c))))
    (uniq-orig sel nc))
  (defgenrule neqRl_indx
    (forall ((x indx)) (implies (fact neq x x) (false))))
  (defgenrule neqRl_strd
    (forall ((x strd)) (implies (fact neq x x) (false))))
  (defgenrule neqRl_mesg
    (forall ((x mesg)) (implies (fact neq x x) (false)))))
```

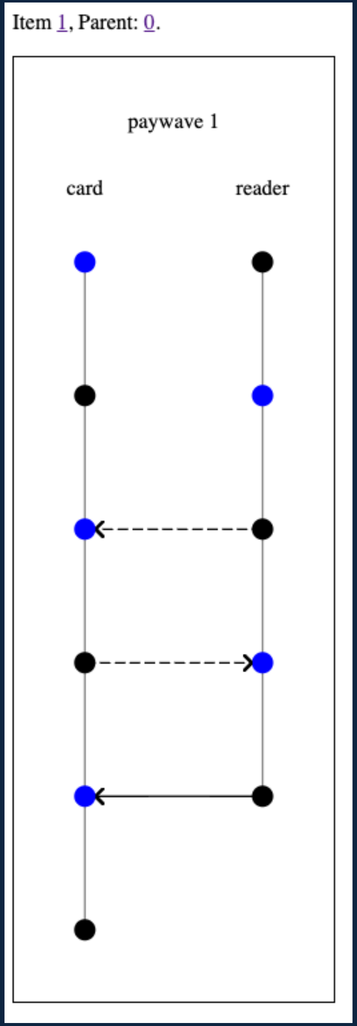


```
(defskeleton paywave
  (vars (opts nr mesg) (sel nc text) (r c name))
  (defstrand card 6 (opts opts) (nr nr) (sel sel) (nc nc) (r r) (c c))
  (non-orig (privk r) (privk c))
  (uniq-orig sel nc)
  (traces
    ((recv (cat r opts)) (send (cat r c sel)) (recv (cat r c nr))
      (send (cat r c nr nc)) (recv (enc nr nc (privk r)))
      (send (enc nr nc (privk c)))))
  (label 0)
  (unrealized (0 4))
  (origs (sel (0 1)) (nc (0 3)))
  (comment "1 in cohort - 1 not yet seen"))
```

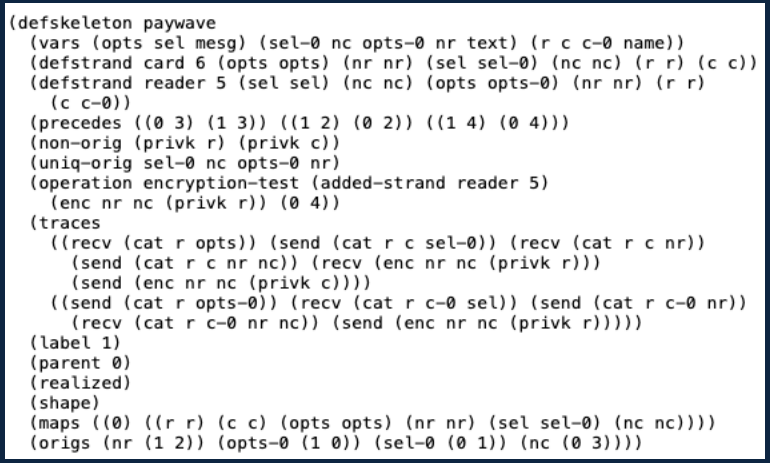
Item 0, Child: 1.



Item 1, Parent: 0.



<u>Some</u> of the information that CPSA gives you for just one query.

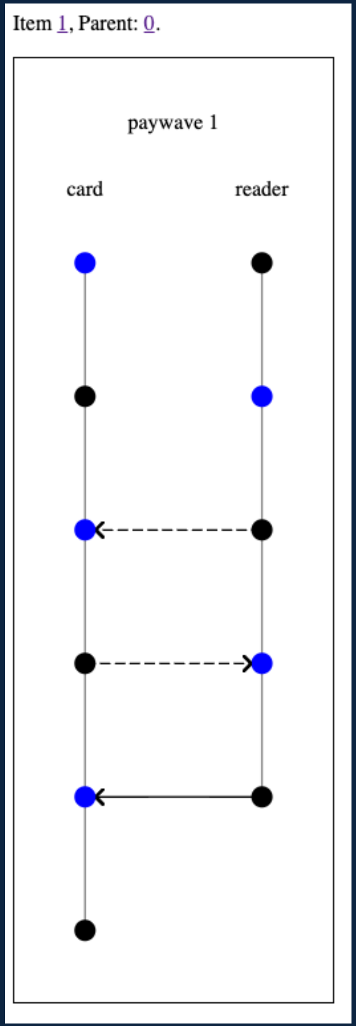


```
(defskeleton paywave
  (vars (opts sel mesg) (sel-0 nc opts-0 nr text) (r c c-0 name))
  (defstrand card 6 (opts opts) (nr nr) (sel sel-0) (nc nc) (r r) (c c))
  (defstrand reader 5 (sel sel) (nc nc) (opts opts-0) (nr nr) (r r)
    (c c-0))
  (precedes ((0 3) (1 3)) ((1 2) (0 2)) ((1 4) (0 4)))
  (non-orig (privk r) (privk c))
  (uniq-orig sel-0 nc opts-0 nr)
  (operation encryption-test (added-strand reader 5)
    (enc nr nc (privk r)) (0 4))
  (traces
    ((recv (cat r opts)) (send (cat r c sel-0)) (recv (cat r c nr))
      (send (cat r c nr nc)) (recv (enc nr nc (privk r)))
      (send (enc nr nc (privk c))))
    ((send (cat r opts-0)) (recv (cat r c-0 sel)) (send (cat r c-0 nr))
      (recv (cat r c-0 nr nc)) (send (enc nr nc (privk r)))))
  (label 1)
  (parent 0)
  (realized)
  (shape)
  (maps ((0) ((r r) (c c) (opts opts) (nr nr) (sel sel-0) (nc nc))))
  (origs (nr (1 2)) (opts-0 (1 0)) (sel-0 (0 1)) (nc (0 3))))
```

What information should I focus on?

MITRE

Translation Goal



GOAL #1:
Highlight the most relevant information.

Security Goal:
Authentication from the Card's pov.

Must Exist Roles:
Reader

Authenticated Messages:
Card Mesg 5 == Reader Mesg 5

Potential Issues
Mesg 1-4 are not authenticated.

GOAL #2:
Provide a suggestion if security goal is not fully satisfied.

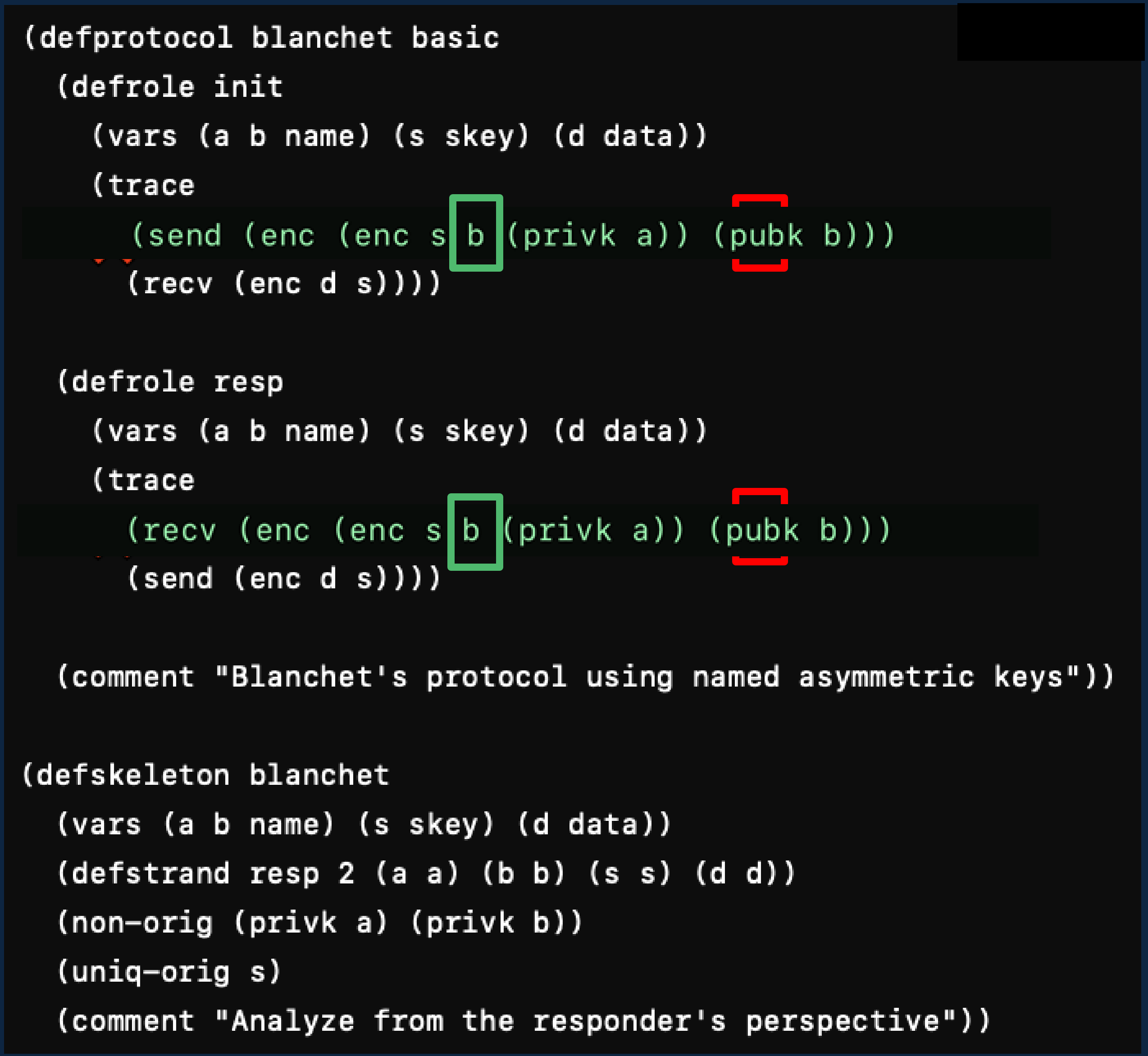
Suggestion:

Add the value 'c' to
Card Mesg 5 and Reader Mesg 5.

Don't need a LLM

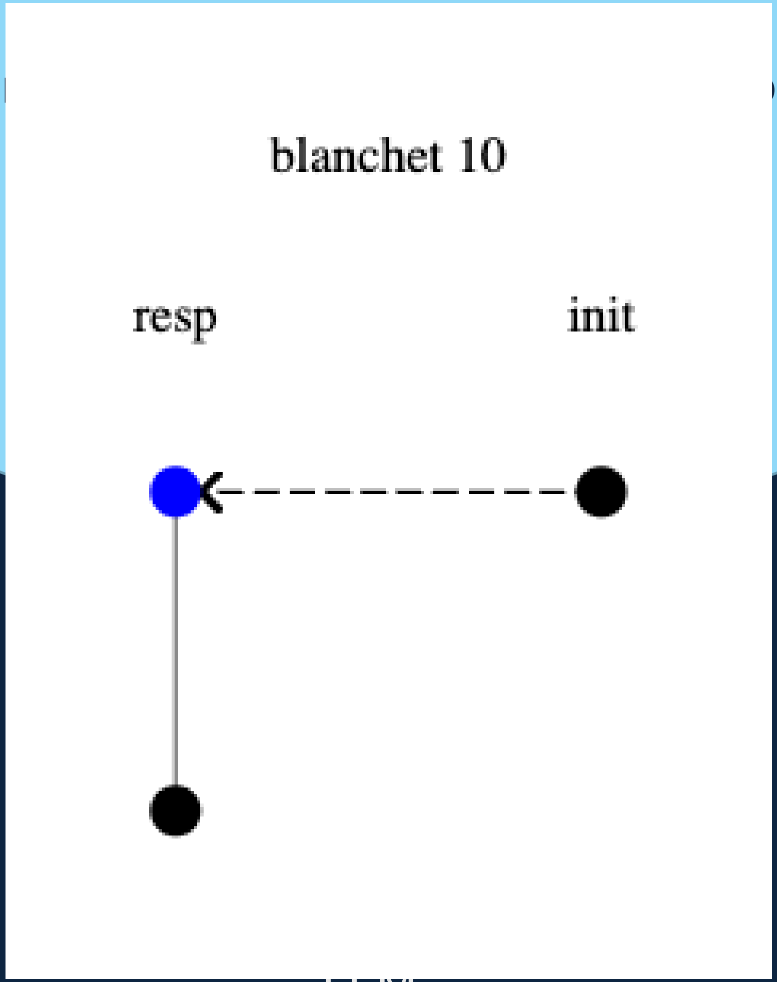
Parsing Script

Translation Goal



```
(defprotocol blanchet basic
  (defrole init
    (vars (a b name) (s skey) (d data))
    (trace
      (send (enc (enc s b (privk a)) (pubk b)))
      (recv (enc d s))))

  (defrole resp
    (vars (a b name) (s skey) (d data))
    (trace
      (recv (enc (enc s b (privk a)) (pubk b)))
      (send (enc d s))))

  (comment "Blanchet's protocol using named asymmetric keys"))

(defskeleton blanchet
  (vars (a b name) (s skey) (d data))
  (defstrand resp 2 (a a) (b b) (s s) (d d))
  (non-orig (privk a) (privk b))
  (uniq-orig s)
  (comment "Analyze from the responder's perspective"))
```

ANALYZE IN CPSA



LLM

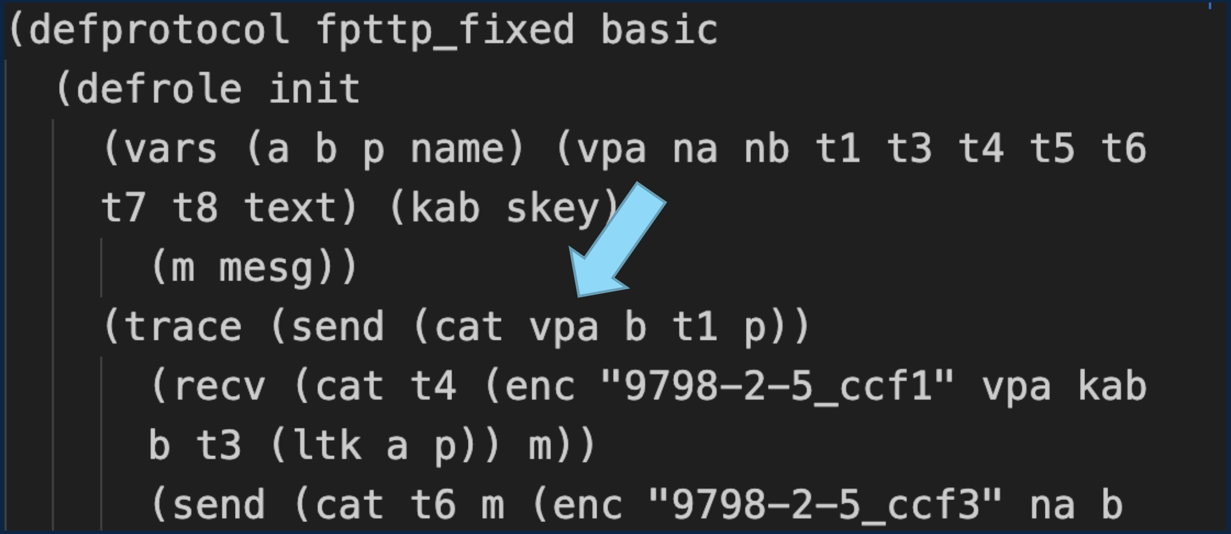
Main Conclusions

Improving the usability for CPSA.

Combing formal methods with LLMs.

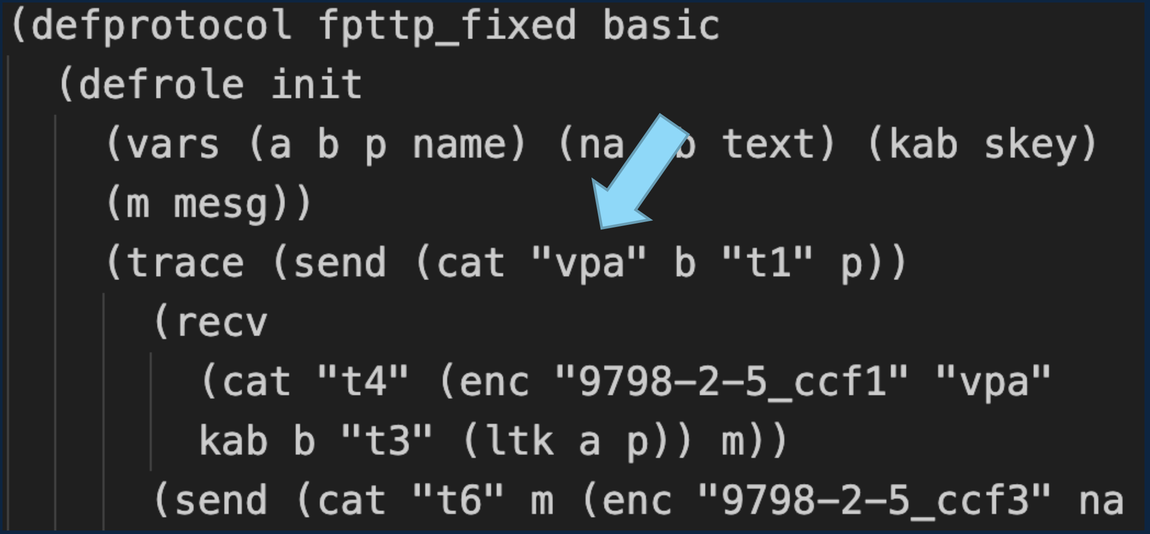
- **NL → CPSA:** LLMs can reliably serve as translators between NL and CPSA, achieving 98.3% accuracy.
- **CPSA → NL:** LLMs can act as assistants, working in tandem with a CPSA analysis parser to both identify issues in protocols and strengthen them.
- **Future:** More complex protocols, goal translation, CPSA → NL results, integration

Actual vs Generated: 2_5 2.scm



```
(defprotocol fpttp_fixed basic
  (defrole init
    (vars (a b p name) (vpa na nb t1 t3 t4 t5 t6
    t7 t8 text) (kab skey)
      (m mesg))
    (trace (send (cat vpa b t1 p))
      (recv (cat t4 (enc "9798-2-5_ccf1" vpa kab
      b t3 (ltk a p)) m))
      (send (cat t6 m (enc "9798-2-5_ccf3" na b
```

Actual CPSA



```
(defprotocol fpttp_fixed basic
  (defrole init
    (vars (a b p name) (na nb text) (kab skey)
    (m mesg))
    (trace (send (cat "vpa" b "t1" p))
      (recv
        (cat "t4" (enc "9798-2-5_ccf1" "vpa"
        kab b "t3" (ltk a p)) m))
      (send (cat "t6" m (enc "9798-2-5_ccf3" na
```

Generated CPSA

In CPSA:
vpa != "vpa"

Modify Context

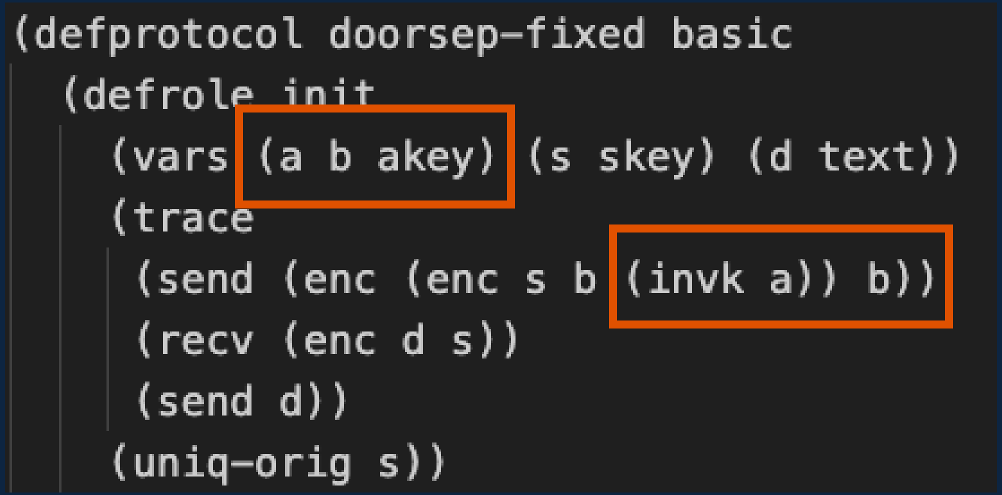
MITRE

Actual vs Generated: doorsep 5.scm

Issue: Incorrect Encryption Key Notations

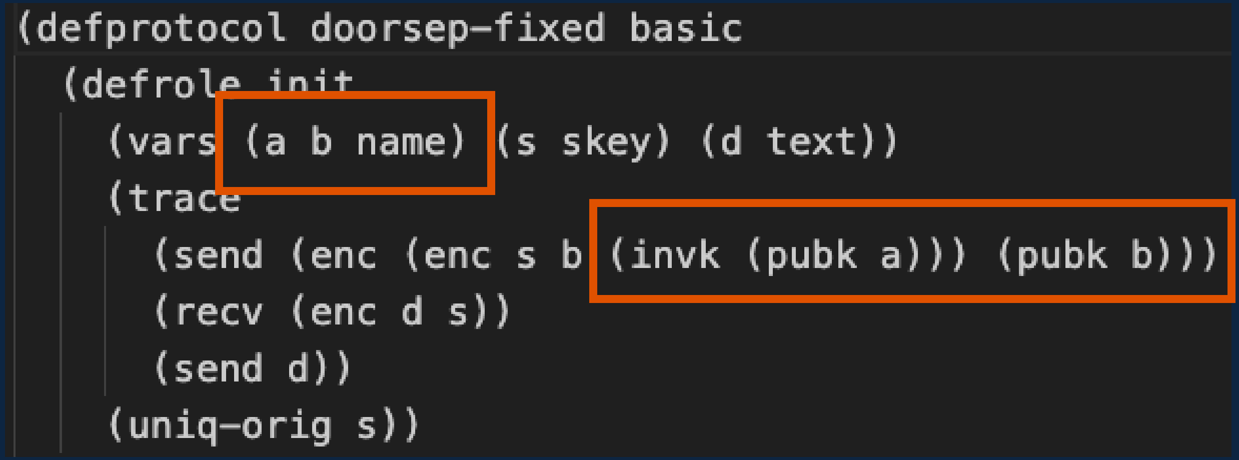
Allowed CPSA formats:
(a, invk a) or (pubk a, privk b)
BUT
Context examples only show
(pubk a, privk b) form

Modify Context Examples



```
(defprotocol doorsep-fixed basic
  (defrole init
    (vars (a b akey) (s skey) (d text))
    (trace
     (send (enc (enc s b (invk a)) b))
     (recv (enc d s))
     (send d))
    (uniq-orig s))
```

Actual CPSA



```
(defprotocol doorsep-fixed basic
  (defrole init
    (vars (a b name) (s skey) (d text))
    (trace
     (send (enc (enc s b (invk (pubk a))) (pubk b)))
     (recv (enc d s))
     (send d))
    (uniq-orig s))
```

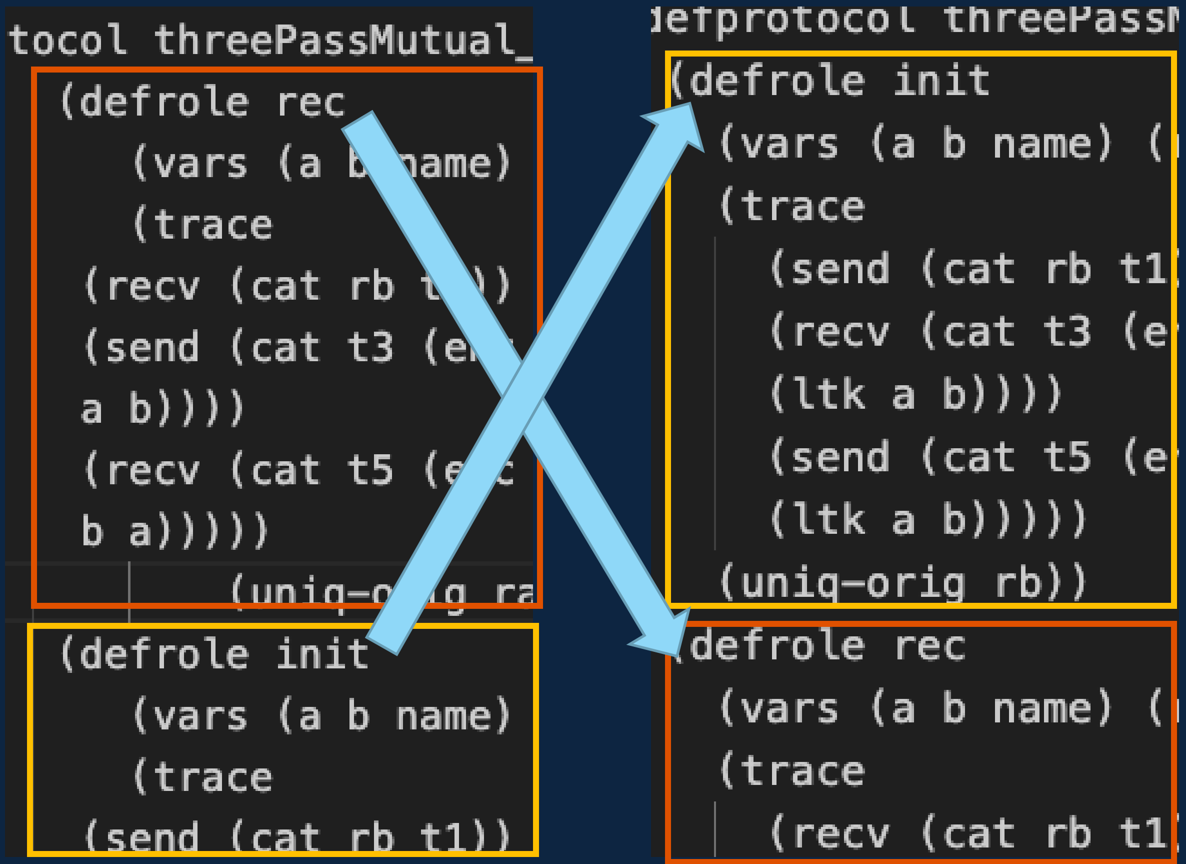
Generated CPSA

Actual vs Generated: 2-4 2.scm

<u>Issue:</u> Roles Switched

Logically Equivalent in CPSA
(CPSA doesn't care about the order of the roles)
BUT
Not in the Scoring Method
(Scoring Method cares about the ordering)

Modify Scoring Method



```
tocol threePassMutual_
  (defrole rec
     (vars (a b name)
     (trace
   (recv (cat rb t1))
   (send (cat t3 (e
   a b))))
   (recv (cat t5 (e  c
   b a)))))
          (uniq-orig ra
  (defrole init
     (vars (a b name)
     (trace
   (send (cat rb t1))
```

```
defprotocol threePassM
(defrole init
  (vars (a b name) (
  (trace
    (send (cat rb t1
    (recv (cat t3 (e
    (ltk a b))))
    (send (cat t5 (e
    (ltk a b)))))
  (uniq-orig rb))
(defrole rec
  (vars (a b name) (
  (trace
    (recv (cat rb t1
```

Actual CPSA **Generated CPSA**