Protocol Composition Logic

Arnab Roy

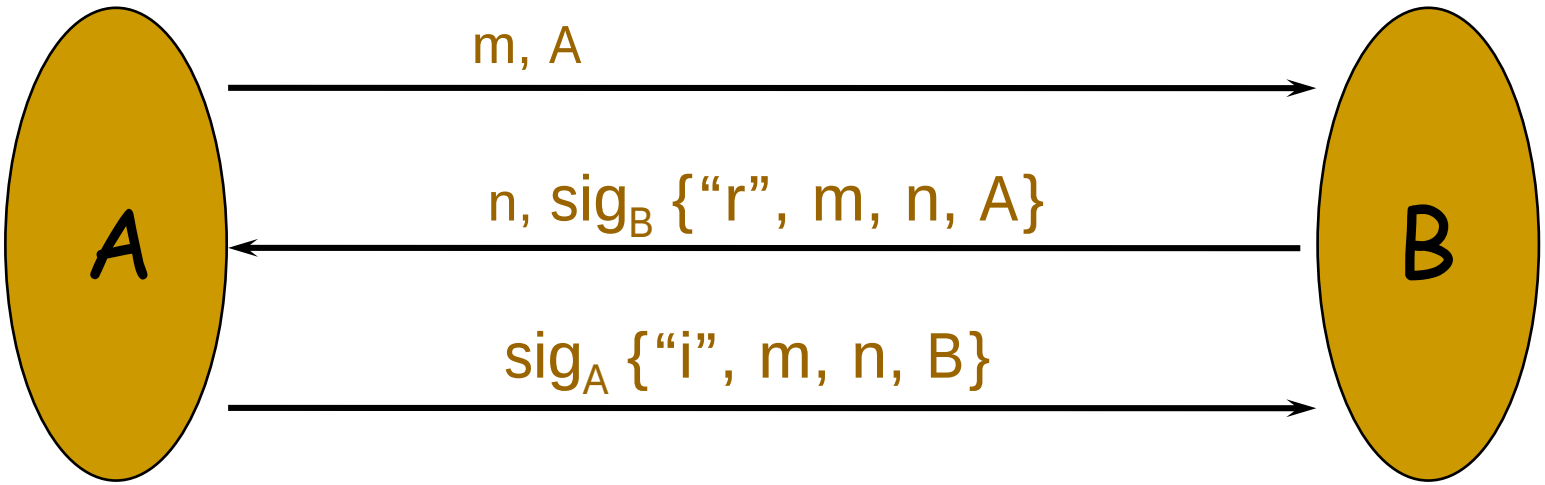
joint work with

A. Datta, A. Derek, N. Durgin, J.C. Mitchell, D. Pavlovic

Today's Plan

- First half
  - The meaning, importance and technique of *proving* protocols secure
  - Our approach: Protocol Composition Logic (PCL)
- Second half
  - Mukund is going to talk about proving IEEE 802.11i secure

Challenge-Response Protocol



Matching Conversation for B

- If B completes protocol

  Then

  B sent msg1 before A received msg1 and

  A received msg1 before A sent msg2 and

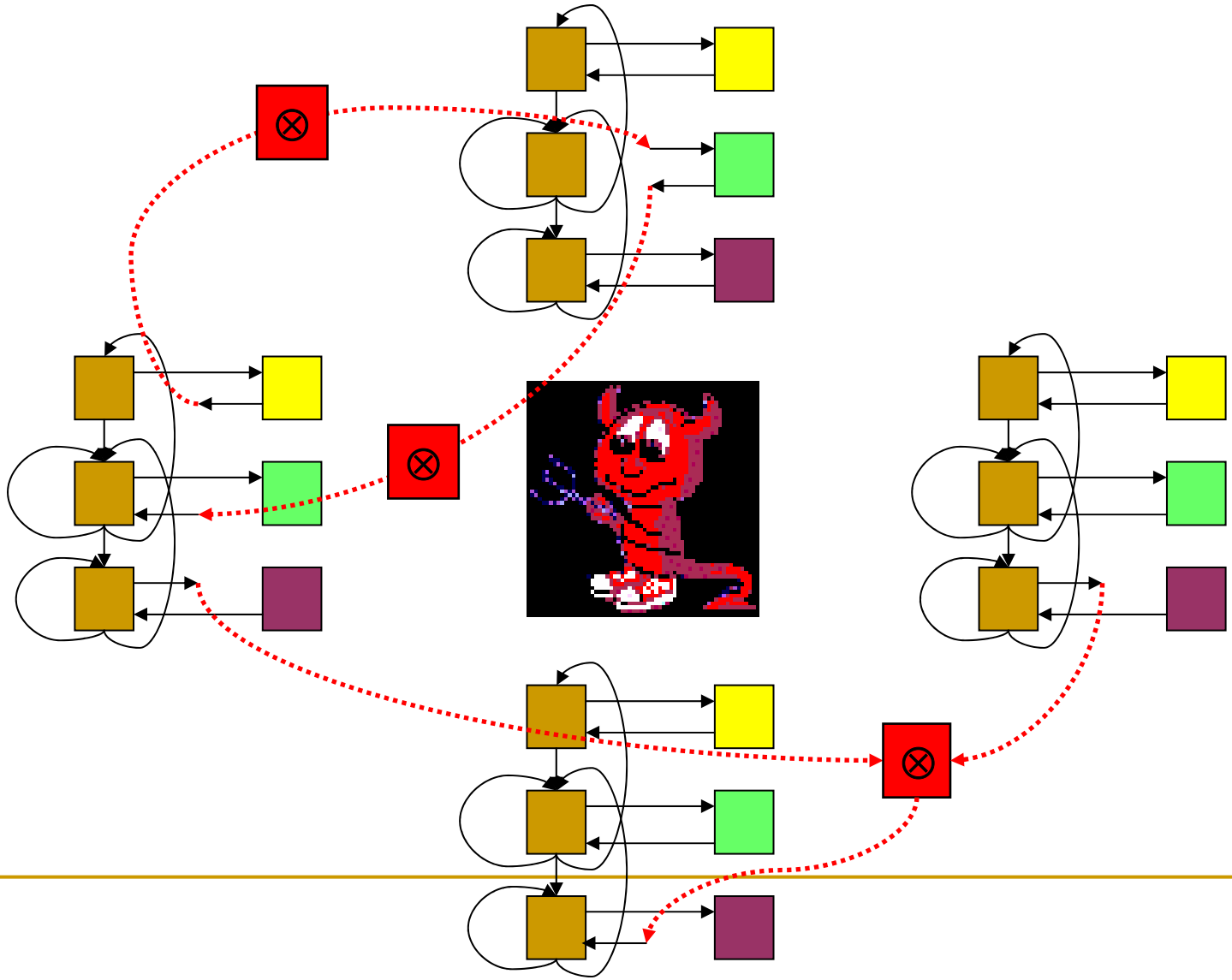
  A sent msg2 before B received msg2 and

  B received msg2 before B sent msg3

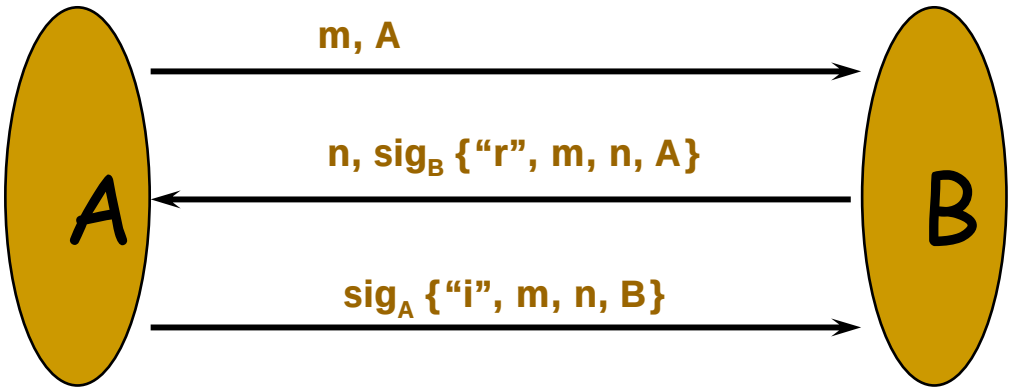
Symbolic Model

- Assume Perfect Cryptography
  - Perfect Encryptions – cannot be decrypted without decryption key
  - Unforgeable Signatures – cannot be produced without signing key
  - Unguessable Nonces
- Attacker can
  - Concatenate messages
  - Unpair concatenations
  - Encrypt, Decrypt, Sign with known keys
  - Generate own nonces

General Active Attack Scenario



Proof Idea

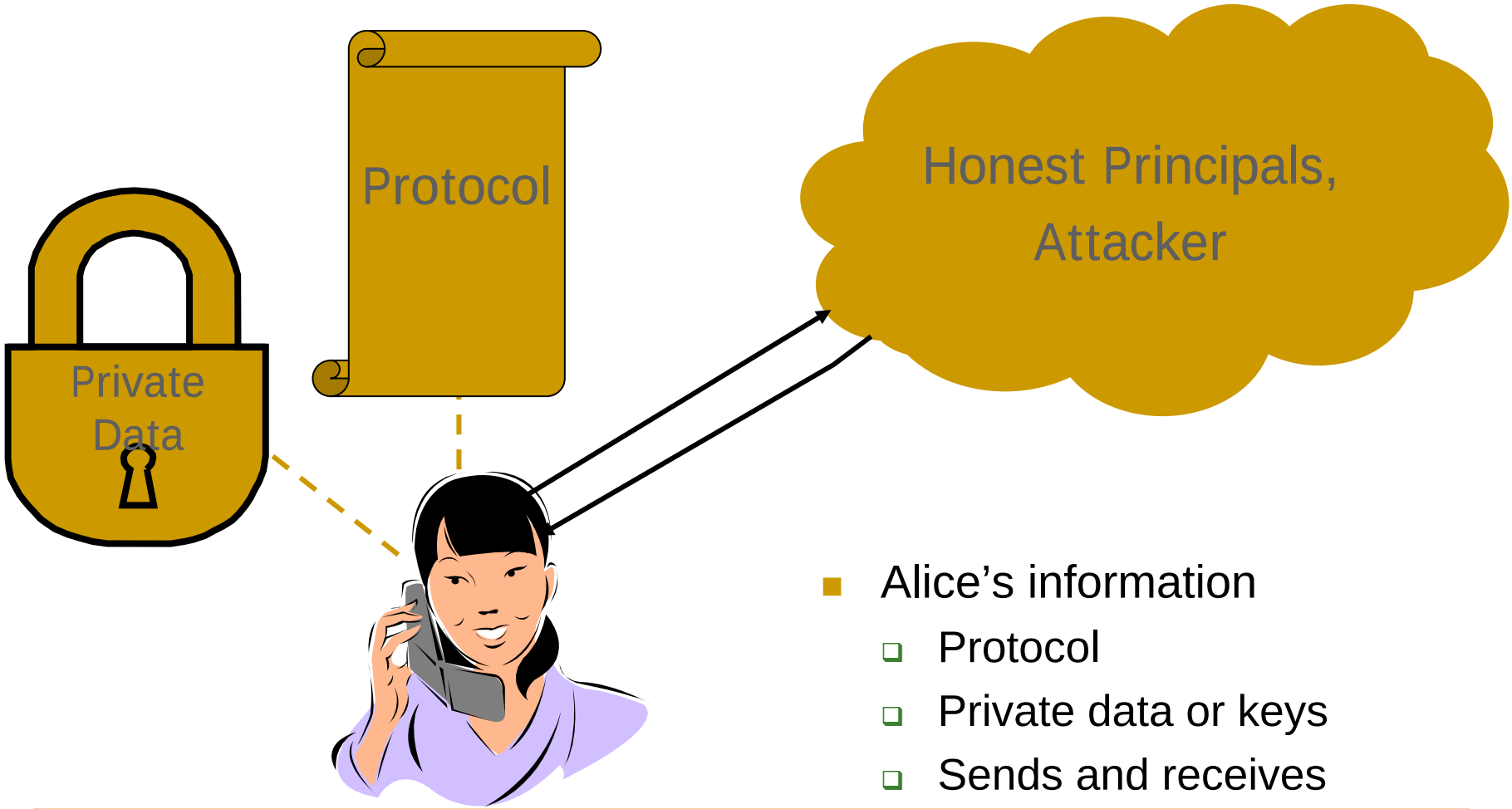


1. B received A’s signature sig$_A$ {“i”, m, n, B} – so A must have signed it.	Property of signatures
2. A must have received the msg n, sig$_B$ {“r”, m, n, A}	Property of the protocol
2. And before that A must have sent the msg m, A	Property of the protocol
3. A must have sent msg1 before B received it – freshness of m	Property of nonces
4. B must have sent msg2 before A received it – freshness of n	Property of nonces
5. A must have sent msg3 after receiving msg2	Property of the protocol

Protocol Composition Logic: PCL

- Intuition
- Formalism
  - Protocol programming language
  - Protocol logic
    - Syntax
    - Semantics
  - Proof System
- Example
  - Signature-based challenge-response

PCL - Intuition



- Alice's information
  - Protocol
  - Private data or keys
  - Sends and receives

Logic: Background

- Logic
  - Syntax — Formulas
    - p, $p \vee q$, $\neg(p \vee q)$, $p \Rightarrow q$
  - Semantics — Truth
    - Model, M = {p = true, q = false}

      $M \models p \vee q$
- Proof System
  - Axioms and proof rules — Provability
    - $p \Rightarrow (q \Rightarrow p)$

      $$\frac{p \quad p \Rightarrow q}{q}$$
  - Soundness Theorem
    - Provability implies truth
    - Axioms and proof rules hold in all “relevant” models

Actions

send t;	send a term t
receive x;	receive a term into variable x
new n;	generate nonce n

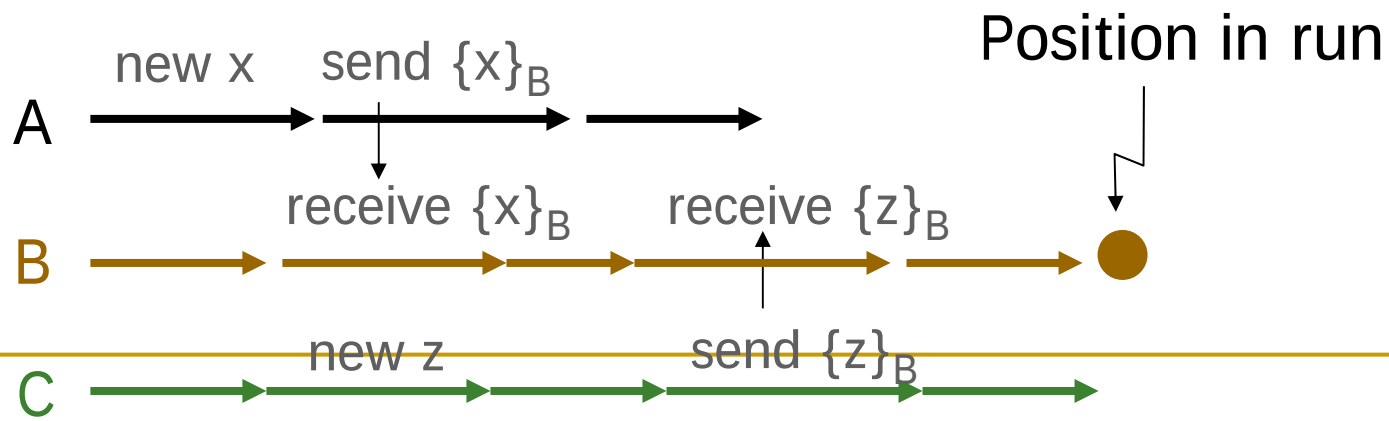
- A program is just a sequence of actions

InitCR(A, X) = [
new m;
send A, X, {m, A};
receive X, A, {x, sig_X{“r”, m, x, A}};
send A, X, sig_A{“i”, m, x, X}};
]$_A$

RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, sig_B{“r”, y, n, Y}};
receive Y, B, sig_Y{“i”, y, n, B}};
]$_B$

Execution Model

- Initial Configuration, IC
  - Set of principals and keys
  - Assignment of ≥ 1 role to each principal
- Run
  - Interleaving of actions of honest principals and attacker starting from IC



Formulas true at a position in run

- Action formulas

  a ::= Send(P,t) | Receive (P,t) | New(P,t)
  | Decrypt (P,t) | Verify (P,t)

- Formulas

  φ ::= a | Has(P,t) | Fresh(P,t) | Honest(N)
  | Contains(t_1, t_2) | $\neg\varphi$ | $\varphi_1 \wedge \varphi_2$ | $\exists x\ \varphi$
  | a < a

- Modal formula

  φ [*actions*] $_P$ φ

- Example

  Has(X, secret) $\supset$ (X = A $\vee$ X = B)

Specifying secrecy

Semantics

- Protocol Q
  - Defines set of roles (e.g., initiator, responder)
  - Run R of Q is sequence of actions by principals following roles, plus attacker
- Satisfaction
  - Q, R $\models \theta$ [*actions*] $_P \varphi$

    If some role of P in R does exactly *actions* starting from state where θ is true, then φ is true in state after *actions* completed
  - Q $\models \theta$ [*actions*] $_P \varphi$

    Q, R $\models \theta$ [*actions*] $_P \varphi$ for all runs R of Q

Challenge-Response Property

- Specifying authentication for Responder

$$
\begin{aligned}
&CR \models \text{true}\ [\ RespCR(A)\ ]_B\ Honest(A) \supset (\\
&\quad Send(A, \{A,B,m\}) < Receive(B, \{A,B,m\}) \wedge\\
&\quad Receive(B, \{A,B,m\}) < Send(B, \{B,A,\{n, sig_B\ \{\text{“r”},m, n, A\}\}\}) \wedge\\
&\quad Send(B, \{B,A,\{n, sig_B\ \{\text{“r”},m, n, A\}\}\}) < Receive(A, \{B,A,\{n, sig_B\ \{\text{“r”},m, n, A\}\}\}) \wedge\\
&\quad Receive(A, \{B,A,\{n, sig_B\ \{\text{“r”},m, n, A\}\}\}) < Send(A, \{A,B,\{sig_A\{\text{“i”},m,n,B\}\}\}) \wedge\\
&\quad Send(A, \{A,B,\{sig_A\{\text{“i”},m,n,B\}\}\} < Receive(B, \{A,B,\{sig_A\{\text{“i”},m,n,B\}\}\})\ )\\
&\quad)
\end{aligned}
$$

Authentication as “matching conversations” [Bellare-Rogaway93]

Proof System

- Goal: Formally prove security properties
- Axioms
  - Simple formulas provable by hand
- Inference rules
  - Proof steps
- Theorem
  - Formula obtained from axioms by application of inference rules

Sample axioms

- Actions

  true [send m]$_P$ Send(P,m)

$$\mathbf{AA4} \qquad \top[a;\cdots;b]x\,\mathsf{a}(\mathsf{b}$$

- Nonce freshness

$$\mathbf{FS2} \qquad \mathsf{FirstSend}(X,t,t') \wedge \mathsf{a}(Y,t'') \supset \mathsf{Send}(X,t')\langle\mathsf{a}(Y,t'')$$
$$\text{where } X \neq Y \text{ and } t \subseteq t''.$$

Encryption and signature

- Public key encryption

  $\text{Honest}(X) \wedge \text{Decrypt}(Y, \text{enc}_X\{m\}) \supset X=Y$

- Signature

  $\text{Honest}(X) \wedge \text{Verify}(Y, \text{sig}_X\{m\}) \supset \text{Sign}(X, \text{sig}_X\{m\})$

Correctness of CR – step 1

InitCR(A, X) = [
new m;
send A, X, {m, A};
receive X, A, {x, sig_X{"r", m, x, A}};
send A, X, sig_A{"i", m, x, X}};
]$_A$

RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, sig_B{"r", y, n, Y}};
receive Y, B, sig_Y{"i", y, n, B}};
]$_B$

1. B reasons about his own action

CR |- true [RespCR(B)] $_B$ Verify(B, sig_A {"i", m, n, A})

2. Use signature axiom

CR |- true [RespCR(B)] $_B$ Sign(A, sig_A{"i", m, n, A})

Proving Invariants

- We want to prove
  - $\Gamma \equiv \text{Honest}(X) \rightarrow \varphi$,

    where

    $\varphi \equiv (\text{Sign}(X, \text{sig}_X(\text{“i”}, m, n, Y) \rightarrow \text{Receive}(Y, n, \text{sig}_Y(\text{“r”}, m, n, X)))$
- Invariant holds if \phi holds at all pausing states of all traces.
  - Since the fragment of honest party action between pausing states is a protocol segment, the propagation of φ looks like:
  - φ --- actions of A --- φ ---- actions of B --- φ --- attacker actions -- φ ---- actions of B --- φ -- …

Proving Invariants (2)

- This gives the following rule for establishing Γ:
  - Prove φ holds when threads have started.
  - Prove, for all protocol segments, if φ held at the beginning, it holds at the end.

Proving Invariants (3)

- Consider the protocol segments of CR
  - For all protocol segments except Init2, Sign(X, sig_X(“i”, m, n, Y)) is false – so φ holds trivially.
  - For Init2, Sign(X, sig_X(“i”, m, n, Y)) and Receive(Y, n, sig_Y(“r”, m, n, X)) both hold – so φ holds again.
- Hence Γ holds!

InitCR(A, X) = [

new m;

send A, X, {m, A};

receive X, A, {x, sig_X{“r”, m, x, A}};

send A, X, sig_A{“i”, m, x, X}};

]$_A$

RespCR(B) = [

receive Y, B, {y, Y};

new n;

send B, Y, {n, sig_B{“r”, y, n, Y}};

receive Y, B, sig_Y{“i”, y, n, B}};

]$_B$

Correctness of CR – step 2

- So far
  - CR |- true [RespCR(B)]$_B$ Sign(A, sig$_A${“i”, m, n, A})
- Apply Γ to prove:
  - CR |- true [RespCR(B)]$_B$ Receive(A, n, sigB{“r”, m, n, A})
- Reason from B’s point of view to prove:
  - CR |- true [RespCR(B)]$_B$ FirstSend(B, n, (n, sigB{“r”, m, n, A}))
- Apply Nonce freshness axiom to prove:
  - CR |- true [RespCR(B)]$_B$ Receive(A, (n, sigB{“r”, m, n, A})) < Send(B, sigB{“r”, m, n, A})
- A few similar steps leads to the full proof!

Thanks!

and over to Mukund