CS 395T

Compositional Protocol Logic

Outline

- Floyd-Hoare logic of programs
  - Compositional reasoning about properties of programs
- DDMP protocol logic
  - Developed by Datta, Derek, Mitchell, and Pavlovic for logical reasoning about security properties

Floyd-Hoare Logic

- Main idea: before-after assertions
  - F <P> G
    - If F is true before executing P, then G is true after
- Total correctness or partial correctness
  - Total correctness: F [P] G
    - If F is true, then P will halt and G will be true
  - Partial correctness: F {P} G
    - If F is true and if P halts, then G will be true

While Programs

P ::= x := e |
P ; P |
if B then P else P |
while B do P

where x is any variable
e is any integer expression
B is a Boolean expression (true or false)

Assignment and Rule of Consequence

- Assignment axiom: $F(t) \ \{ x := t \} \ F(x)$
  - If F holds for t, and t is assigned to x, then F holds for x aftewards
  - This assumes that there is no aliasing!
  - Examples:

    $7=7 \quad \{ x := 7 \} \quad x=7$

    $(y+1)>0 \quad \{ x := y+1 \} \quad x>0$

    $x+1=2 \quad \{ x := x+1 \} \quad x=2$

- Rule of consequence:

  If $F \ \{ P \} \ G$ and $F' \to F$ and $G \to G'$,

  then $F' \ \{ P \} \ G'$

Simple Examples

- Assertion: $y>0 \ \{ x := y+1 \} \ x>0$

  Proof:

  $(y+1)>0 \quad \{ x := y+1 \} \quad x>0$ (assignment axiom)

  $y>0 \quad \{ x := y+1 \} \quad x>0$ (rule of consequence)

  $y>0 \rightarrow y+1>0$

- Assertion: $x=1 \ \{ x := x+1 \} \ x=2$

  Proof:

  $x+1=2 \quad \{ x := x+1 \} \quad x=2$ (assignment axiom)

  $x=1 \quad \{ x := x+1 \} \quad x=2$ (rule of consequence)

Conditional

$$\frac{F \,\&\, B \ \{\ P\ \}\ G \qquad F \,\&\, \neg B\ \{\ Q\ \}\ G}{F\ \{\ \text{if } B \text{ then } P \text{ else } Q\ \}\ G}$$

- Example:

$\text{true} \ \{ \text{ if } y \geq 0 \text{ then } x := y \text{ else } x := -y \ \} \ x \geq 0$

Sequence

$$\frac{F \;\{P\}\; G \qquad G \;\{Q\}\; H}{F \;\{P;Q\}\; H}$$

- Example:

$x{=}0 \;\{\, x := x{+}1 \,;\; x := x{+}1 \,\}\; x{=}2$

Loop Invariant

$$\frac{F \,\&\, B \;\{\; P \;\}\; F}{F \;\{\; \text{while } B \text{ do } P \;\}\; F \,\&\, \neg B}$$

F is the loop invariant; it should hold before and after the loop body

- Example:

true { while $x \neq 0$ do x := x-1 } x=0

Example: Compute d=x-y

- **Assertion:** $y \le x \ \{\underbrace{d:=0}_{P};\ \text{while}\ \underbrace{(y+d)<x}_{B}\ \text{do}\ \underbrace{d := d+1}_{Q}\}\ y+d=x$
- **Proof:**
  - Choose loop invariant F = $y+d \le x$

$$\frac{y+d \le x \ \&\ B\ \ \{Q\}\ \ y+d \le x}{y+d \le x\ \{\text{while } B \text{ do } Q\}\ y+d \le x \ \&\ \neg B}$$

After loop execution,
$y+d \le x \ \& \neg(y+d<x)$,
thus $y+d=x$

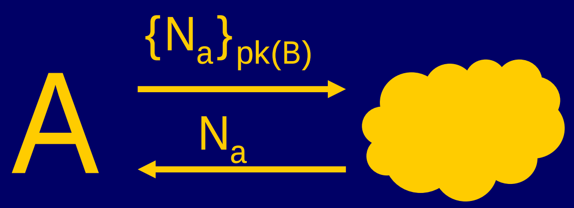
    - Important: proving a property of the entire loop has been reduced to proving a property of one iteration of the loop
  - To prove $y+d \le x \ \&\ B\ \ \{Q\}\ \ y+d \le x$, use assignment axiom and sequence rule

Goal: Logic for Security Protocols

- “Floyd-Hoare” reasoning about security properties
  - Would like to derive <u>global</u> properties of protocols from <u>local</u> assertions about each protocol participant
  - Use a rigorous logical framework to formalize the reasoning that each participant carries out
- Compositionality is important
  - Security properties must hold even if the protocol is executed in parallel with other protocols
  - Compositionality is the main advantage of process calculi and protocol logics

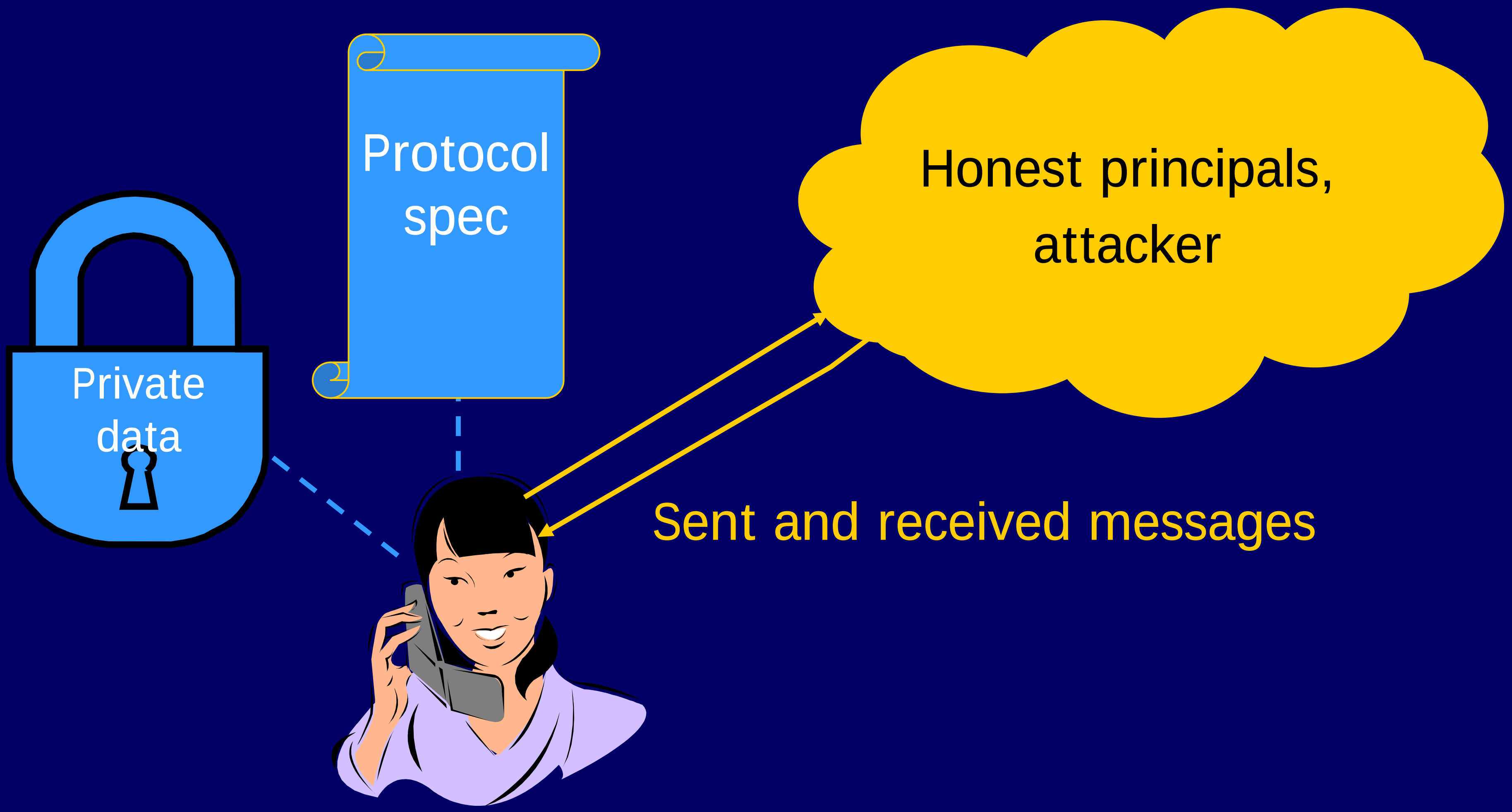
Intuition

- Reason about local information
  - I chose a fresh, unpredictable number
  - I sent it out encrypted
  - I received it decrypted
  - Therefore: someone decrypted it

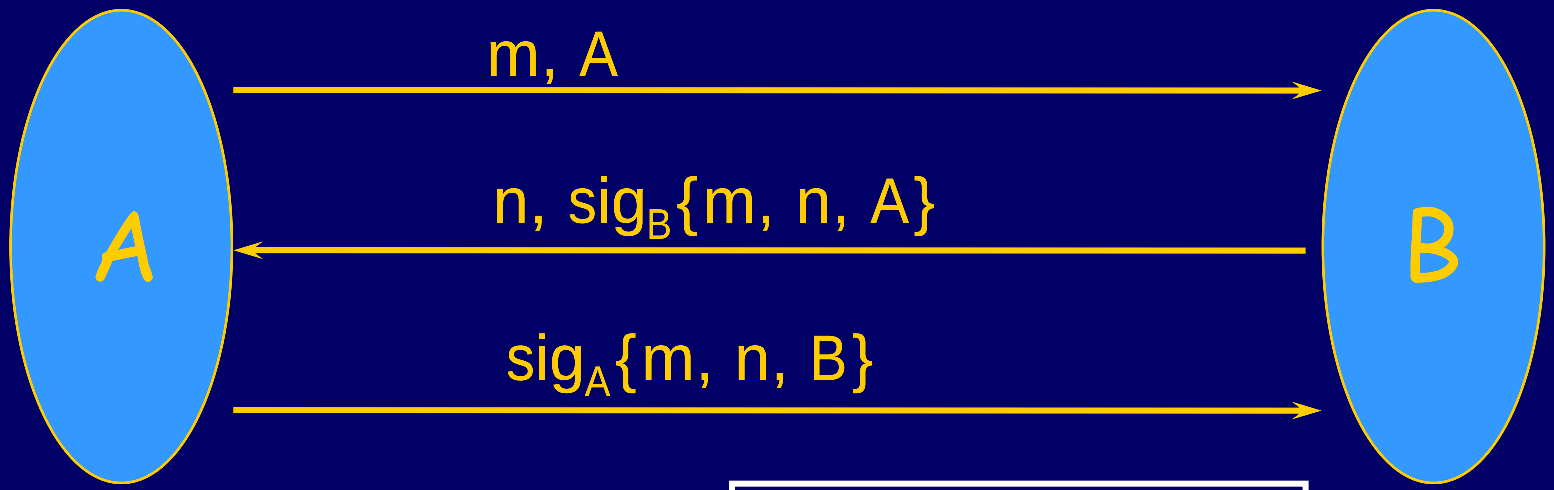


- Incorporate knowledge about protocol into reasoning
  - According to the protocol specification, server only sends m if it received m'
  - If *server not corrupt* and I receive m signed by server, then server received m'

Alice’s “View” of the Protocol



Example: Challenge-Response



protocol-independent reasoning

- Alice's reasoning:
  - If Bob is honest, then only Bob can generate his signature
  - If honest Bob generates a signature of the form $sig_B\{m, n, A\}$, then
  1. Bob must have received m, A from Alice
  2. Bob sent $sig_B\{m, n, A\}$ as part of his 2^{nd} message

protocol-specific reasoning

- Alice concludes:
  - Received(B,msg1) & Sent(B,msg2)

Protocol Composition Logic

[Datta et al.]

- A formal language for describing protocols
  - Terms and actions instead of informal arrows-and-messages notation
- Operational semantics
  - Describe how the protocol executes
- Protocol logic
  - State security properties (in particular, secrecy and authentication)
- Proof system
  - Axioms and inference rules for formally proving security properties

Terms

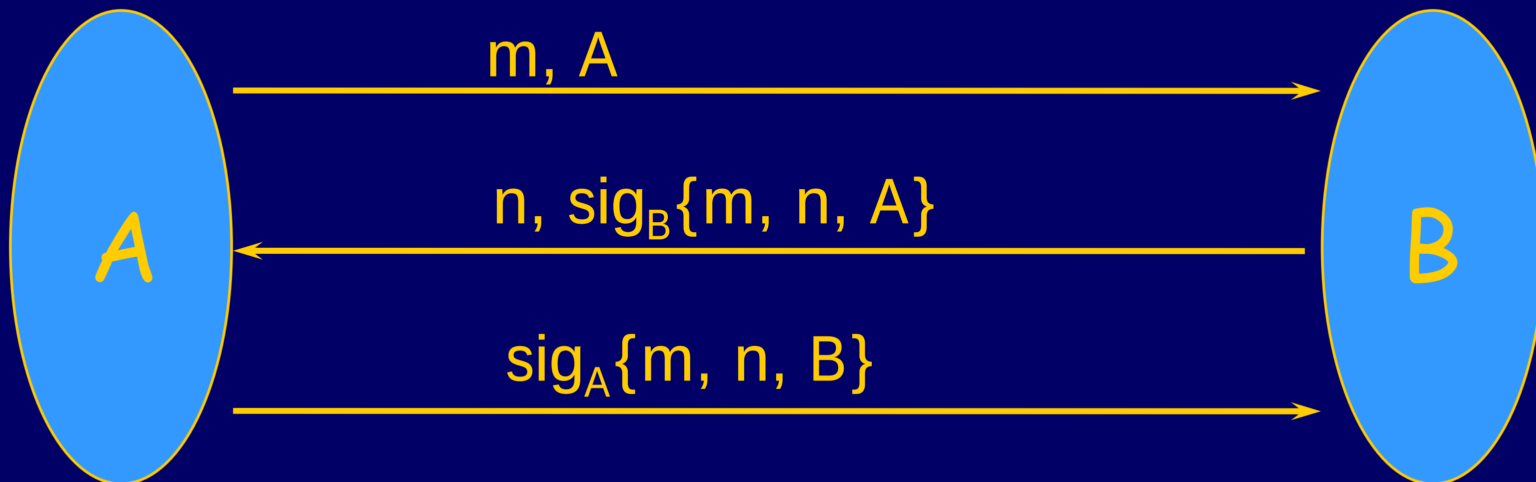
$t ::=$	$c \mid$	constant
	$x \mid$	variable
	$N \mid$	name
	$K \mid$	key
	$t, t \mid$	tuple
	$sig_K\{t\} \mid$	signature
	$enc_K\{t\}$	encryption

Actions

new m	generate fresh value
send U, V, t	send term t from U to V
receive U, V, x	receive term and assign into variable x
match t/p(x)	match term t against pattern p(x)

- A thread is a sequence of actions
  - Defines the “program” for a protocol participant, i.e., what messages he sends and receives and the checks he performs
  - For notational convenience, omit “match” actions
    - Write “receive $sig_B\{A, n\}$” instead of “receive x; match $x/sig_B\{A, n\}$”

Challenge-Response Threads

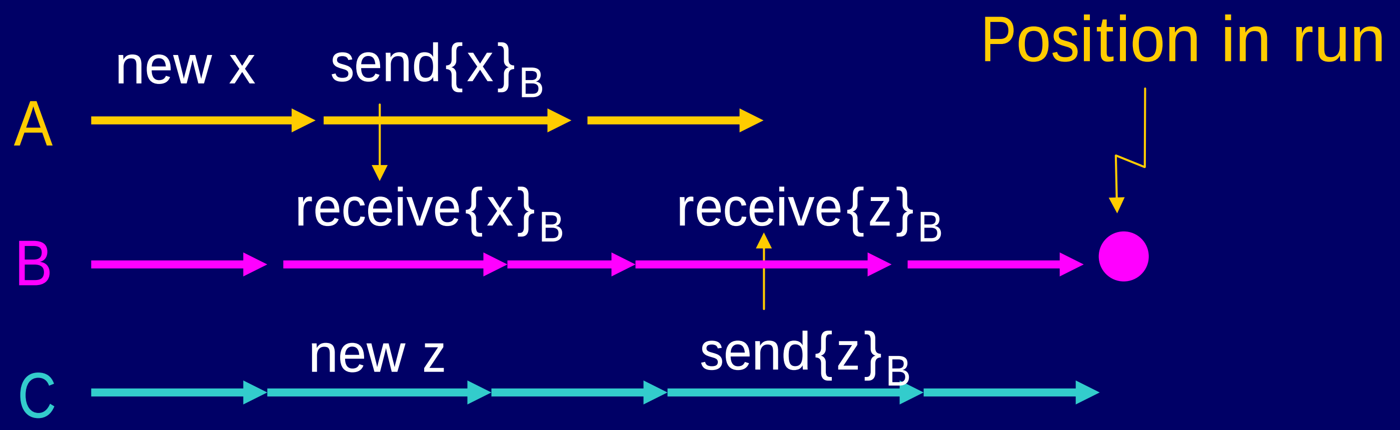


InitCR(A, X) = [
  new m;
  send A, X, {m, A};
  receive X, A, {x, $sig_X\{m, x, A\}$};
  send A, X, $sig_A\{m, x, X\}$;
]

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

Execution Model

- A protocol is a finite set of roles
  - Initial configuration specifies a set of principals and keys; assignment of ≥1 role to each principal
- A run is a concurrent execution of the roles
  - Models a protocol session
  - Send and receive actions are matched up



Action Formulas

- Predicates over action sequences

a ::= Send(X,m)	|	Message m was sent in thread X
Receive(X,m)	|	Message m was received in thread X
New(X,t)	|	Term t was generated as new in X
Decrypt(X,t)	|	Term t was decrypted in thread X
Verify(X,t)		Term t was verified in X

Formulas

$\varphi ::= a$ |	Action formula
$Has(X,m)$ |	Thread X created m or received a message containing m and has keys to extract m from the message
$Fresh(X,t)$ |	Term t hasn't been "seen" outside X
$Honest(N)$ |	Principal N follows protocol rules in all of its threads
$Contains(t,t')$ |	Term t contains subterm t'
$\neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x\ \varphi \mid$	
$\bigcirc\varphi \mid \Diamond\varphi$	Temporal logic operators on past actions

φ was true

Modal operator $[actions]_X\ \varphi$ — After actions, X reasons φ

Trace Semantics

- Protocol Q
  - Defines a set of roles (e.g., initiator and responder)
- Run R
  - Sequence of actions by principals following protocol roles and the attacker (models a protocol session)
- Satisfaction
  - $Q, R \models [\ \textit{actions}\ ]_P\ \varphi$
    - Some role of principal P in R performs exactly *actions* and φ is true in the state obtained after *actions* complete
  - $Q \models [\ \textit{actions}\ ]_P\ \varphi$
    - $Q, R \models [\ \textit{actions}\ ]_P\ \phi$ for all runs R of Q

Specifying Authentication

- Initiator authentication in Challenge-Response

After initiator executes his program

If B is honest...

CR |= [InitCR(A, B)]$_A$ Honest(B) $\supset$

ActionsInOrder(

Send(A, {A,B,m}),

Receive(B, {A,B,m}),

Send(B, {B,A,{n, sig$_B${m, n, A}}}),

Receive(A, {B,A,{n, sig$_B${m, n, A}}})

)

...then msg sends and receives must have happened in order prescribed by protocol spec

Specifying Secrecy

- Shared secret in key establishment

After initiator executes his program

If B is honest...

$$KE \models [\ \mathrm{InitKE}(A, B)\ ]_A\ \mathrm{Honest}(B) \supset$$
$$(\mathrm{Has}(X, m) \supset X{=}A \vee X{=}B\ )$$

... then if some party X knows secret m, then X can only be either A, or B

Proof System

- Goal: formally prove properties of security protocols
- Axioms are simple formulas
  - Provable by hand
- Inference rules are proof steps
- Theorem is a formula obtained from axioms by application of inference rules

Sample Axioms

- New data
  - $[\ \text{new } x\ ]_P\ \ \text{Has}(P,x)$
  - $[\ \text{new } x\ ]_P\ \ \text{Has}(Y,x) \supset Y=P$
- Acquiring new knowledge
  - $[\ \text{receive } m\ ]_P\ \ \text{Has}(P,m)$
- Performing actions
  - $[\ \text{send } m\ ]_P\ \ \Diamond\text{Send}(P,m)$
  - $[\ \text{match } x/\text{sig}_X\{m\}\ ]_P\ \ \Diamond\text{Verify}(P,m)$

Reasoning About Cryptography

- Pairing
  - $\text{Has}(X, \{m,n\}) \supset \text{Has}(X, m) \wedge \text{Has}(X, n)$
- Symmetric encryption
  - $\text{Has}(X, \text{enc}_K(m)) \wedge \text{Has}(X, K^{-1}) \supset \text{Has}(X, m)$
- Public-key encryption
  - $\text{Honest}(X) \wedge \Diamond\text{Decrypt}(Y, \text{enc}_X\{m\}) \supset X=Y$
- Signatures
  - $\text{Honest}(X) \wedge \Diamond\text{Verify}(Y, \text{sig}_X\{m\}) \supset$
    $\exists\, m' \,(\Diamond\text{Send}(X, m') \wedge \text{Contains}(m', \text{sig}_X\{m\})$

Sample Inference Rules

$$\frac{[\text{ actions }]_P \text{ Has}(X, t)}{[\text{ actions; action }]_P \text{ Has}(X, t)}$$

$$\frac{[\text{ actions }]_P \, \phi \qquad\qquad [\text{ actions }]_P \, \varphi}{[\text{ actions }]_P \, \phi \wedge \varphi}$$

Honesty Rule

$\forall$roles R of Q. $\forall$ initial segments A $\subseteq$ R.

$$\frac{Q \ \ |\text{-} \ \ [\ A\ ]_X \ \phi}{Q \ |\text{-} \ \text{Honest}(X) \supset \phi}$$

- Finitary rule (finite number of premises to choose from)
  - Typical protocol has 2-3 roles, typical role has 1-3 actions
- Example:
  - If Honest(X) $\supset$ (Sent(X,m) $\supset$ Received(X,m')) and Y receives a message from X, then Y can conclude Honest(X) $\supset$ Received(X,m')

Correctness of Challenge-Response

InitCR(A, X) = [
    new m;
    send A, X, {m, A};
    receive X, A, {x, sig_X{m, x, A}};
    send A, X, sig_A{m, x, X};
]

RespCR(B) = [
    receive Y, B, {y, Y};
    new n;
    send B, Y, {n, sig_B{y, n, Y}};
    receive Y, B, sig_Y{y, n, B}};
]

$$\begin{aligned}
&CR \vdash [\ InitCR(A, B)\ ]_A\ Honest(B) \supset ActionsInOrder(\\
&\qquad Send(A, \{A,B,m\}), \\
&\qquad Receive(B, \{A,B,m\}), \\
&\qquad Send(B, \{B,A,\{n, sig_B\ \{m, n, A\}\}\}), \\
&\qquad Receive(A, \{B,A,\{n, sig_B\ \{m, n, A\}\}\}) \\
&)
\end{aligned}$$

1: A Reasons about Own Actions

InitCR(A, X) = [
new m;
send A, X, {m, A};
receive X, A, {x, sig_X{m, x, A}};
send A, X, sig_A{m, x, X};
]

RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, sig_B{y, n, Y}};
receive Y, B, sig_Y{y, n, B};
]

CR |- [InitCR(A, B) $]_A$

◇Verify(A, sig_B{m, n, A})

If A completed a protocol session, it must have verified B's signature at some point

2: Properties of Signatures

InitCR(A, X) = [
    new m;
    send A, X, {m, A};
    receive X, A, {x, sig_X{m, x, A}};
    send A, X, sig_A{m, x, X};
]

RespCR(B) = [
    receive Y, B, {y, Y};
    new n;
    send B, Y, {n, sig_B{y, n, Y}};
    receive Y, B, sig_Y{y, n, B};
]

$$\text{CR} \vdash [\ \text{InitCR}(A, B)\ ]_A\ \text{Honest}(B) \supset$$
$$\exists\, t'\ (\Diamond \text{Send}(B, t') \wedge$$
$$\text{Contains}(t', \text{sig}_B\{m, n, A\})$$

If A completed protocol and B is honest, then B must have sent its signature as part of some message

Honesty Invariant

InitCR(A, X) = [
    new m;
    send A, X, {m, A};
    receive X, A, {x, sig_X{m, x, A}};
    send A, X, sig_A{m, x, X};
]

RespCR(B) = [
    receive Y, B, {y, Y};
    new n;
    send B, Y, {n, sig_B{y, n, Y}};
    receive Y, B, sig_Y{y, n, B};
]

$$
\begin{aligned}
CR \vdash\ & \mathrm{Honest}(X) \wedge \\
& \Diamond \mathrm{Send}(X, t') \wedge \mathrm{Contains}(t', sig_X\{y, x, Y\}) \wedge \\
& \neg \Diamond \mathrm{New}(X, y) \supset \\
& \Diamond \mathrm{Receive}(X, \{Y, X, \{y, Y\}\})
\end{aligned}
$$

Honest responder only sends his signature if he received a properly formed first message of the protocol

This condition disambiguates sig_X(...) sent by responder from sig_A(...) sent by initiator

Reminder: Honesty Rule

$\forall$roles R of Q. $\forall$ initial segments A $\subseteq$ R.

$$\frac{Q \;\vdash\; [\,A\,]_X\,\phi}{Q \;\vdash\; \mathrm{Honest}(X) \supset \phi}$$

3: Use Honesty Rule

InitCR(A, X) = [
    new m;
    send A, X, {m, A};
    receive X, A, {x, sig_X{m, x, A}};
    send A, X, sig_A{m, x, X};
]

RespCR(B) = [
    receive Y, B, {y, Y};
    new n;
    send B, Y, {n, sig_B{y, n, Y}};
    receive Y, B, sig_Y{y, n, B};
]

$$CR \mid\!- \ [\ InitCR(A, B)\ ]_A\ Honest(B) \supset \Diamond Receive(B, \{A,B,\{m,A\}\})$$

If A completed protocol and B is honest, then B must have received A's first message

4: Nonces Imply Temporal Order

InitCR(A, X) = [
  new m;
  send A, X, {m, A};
  receive X, A, {x, sig_X{m, x, A}};
  send A, X, sig_A{m, x, X};
]

RespCR(B) = [
  receive Y, B, {y, Y};
  new n;
  send B, Y, {n, sig_B{y, n, Y}};
  receive Y, B, sig_Y{y, n, B};
]

CR |- [InitCR(A, B)]$_A$ Honest(B) $\supset$

ActionsInOrder(...)

Complete Proof

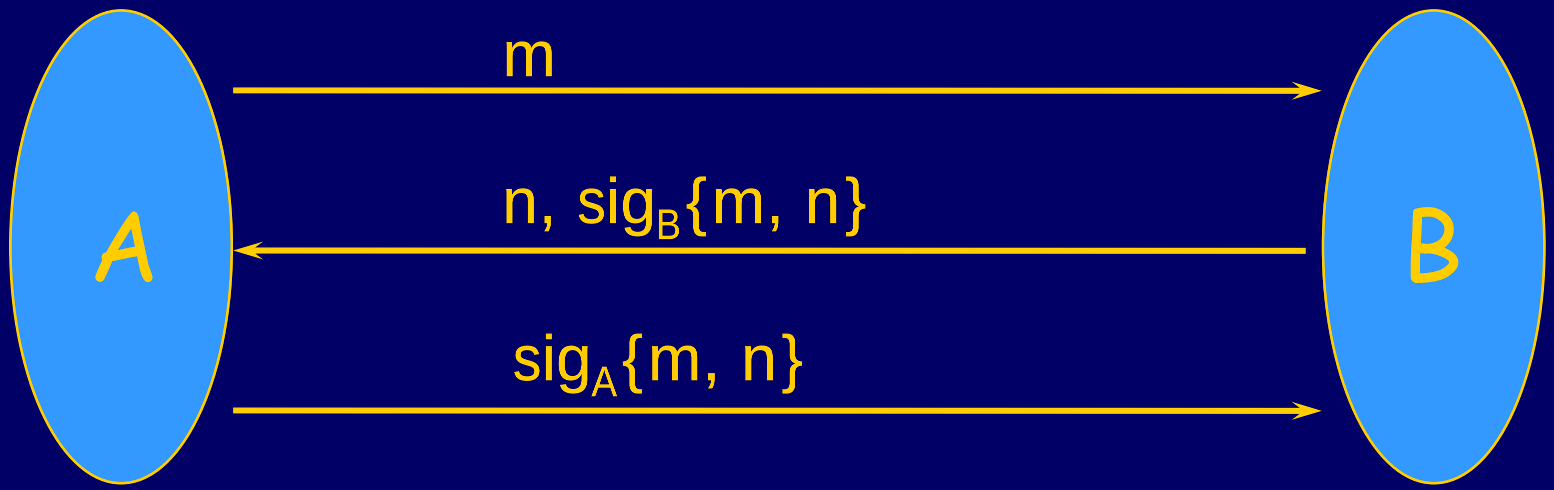
AM1	$(A\ B\ \eta)[\,]_{A,\eta}\ \mathsf{Has}(A, A, \eta) \wedge \mathsf{Has}(A, B, \eta)$
AN3	$[(\nu m)]_{A,\eta}\ \mathsf{Fresh}(A, m, \eta)$
AA1	$[\langle A, B, m\rangle]_{A,\eta}\ \diamondminus\mathsf{Send}(A, \{A, B, m\}, \eta)$
AA1	$[(B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}})]_{A,\eta}$ $\diamondminus\mathsf{Receive}(A, \{B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}, \eta)$
AA1	$[(\{\!\|m, n, A\|\!\}_{\overline{B}} / \{\!\|m, n, A\|\!\}_{B})]_{A,\eta}\ \diamondminus\mathsf{Verify}(A, \{\!\|m, n, A\|\!\}_{\overline{B}}, \eta)$
AA1	$[\langle A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\rangle]_{A,\eta}\ \diamondminus\mathsf{Send}(A, \{A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\}, \eta)$
AF1, AF2	$(A\ B\ \eta)[(\nu m)\langle A, B, m\rangle(x)(x/B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}})$ $(\{\!\|m, n, A\|\!\}_{\overline{B}} / \{\!\|m, n, A\|\!\}_{B})\langle A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\rangle]_{A,\eta}$ $\mathsf{ActionsInOrder}(\mathsf{Send}(A, \{A, B, m\}, \eta), \mathsf{Receive}(A, \{B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}, \eta),$ $\mathsf{Send}(A, \{A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\}, \eta))$
N1	$\diamondminus\mathsf{New}(A, m, \eta) \supset \neg\diamondminus\mathsf{New}(B, m, \eta')$
5, **VER**	$\mathsf{Honest}(B) \wedge \diamondminus\mathsf{Verify}(A, \{\!\|m, n, A\|\!\}_{\overline{B}}, \eta) \supset$ $\exists\eta'.\exists m'.(\diamondminus\mathsf{CSend}(B, m', \eta') \wedge (\{\!\|m, n, A\|\!\}_{\overline{B}} \subseteq m'))$
HON	$\mathsf{Honest}(B) \supset (\exists\eta'.\exists m'.((\diamondminus\mathsf{CSend}(B, m', \eta') \wedge$ $\{\!\|m, n, A\|\!\}_{\overline{B}} \subseteq m' \wedge \neg\diamondminus\mathsf{New}(B, m, \eta')) \supset$ $(\ m' = \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\} \wedge \diamondminus\mathsf{Receive}(B, \{A, B, m\}, \eta') \wedge$ $\mathsf{ActionsInOrder}(\mathsf{Receive}(B, \{A, B, m\}, \eta'), \mathsf{New}(B, n, \eta'),$ $\mathsf{Send}(B, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta')))))$
2, 3, 11, **AF3**	$\mathsf{Honest}(B) \supset \mathsf{After}(\mathsf{Send}(A, \{A, B, m\}, \eta),$ $\mathsf{Receive}(B, \{A, B, m\}, \eta'))$
11, **AF2**	$\mathsf{Honest}(B) \supset \mathsf{After}(\mathsf{Receive}(B, \{A, B, m\}, \eta'),$ $\mathsf{Send}(B, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta'))$
11, 4, **AF3**	$\mathsf{Honest}(B) \supset \mathsf{After}(\mathsf{Send}(B, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta'),$ $\mathsf{Receive}(A, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta))$
$10 - 13$, **AF2**	$\mathsf{Honest}(B) \supset \exists\eta'.(\mathsf{ActionsInOrder}(\mathsf{Send}(A, \{A, B, m\}, \eta),$ $\mathsf{Receive}(B, \{A, B, m\}, \eta'), \mathsf{Send}(B, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta'),$ $\mathsf{Receive}(A, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta))$

Table 8. Deductions of A executing Init role of CR

Properties of Proof System

- Soundness
  - If ϕ is a theorem, then ϕ is a valid formula
    - $Q \mid\!- \phi$ implies $Q \models \phi$
  - Informally: if we can prove something in the logic, then it is actually true
- Proved formula holds in any step of any run
  - There is no bound on the number of sessions!
  - Unlike finite-state checking, the proved property is true for the entire protocol, not for specific session(s)

Weak Challenge-Response



InitWCR(A, X) = [
    new m;
    send A, X, {m};
    receive X, A, {x, $sig_X\{m, x\}$};
    send A, X, $sig_A\{m, x\}$;
]

RespWCR(B) = [
    receive Y, B, {y};
    new n;
    send B, Y, {n, $sig_B\{y, n\}$};
    receive Y, B, $sig_Y\{y, n\}$;
]

1: A Reasons about Own Actions

```
InitWCR(A, X) = [
    new m;
    send A, X, {m};
    receive X, A, {x, sig_X{m, x}};
    send A, X, sig_A{m, x};
]
```

```
RespWCR(B) = [
    receive Y, B, {y};
    new n;
    send B, Y, {n, sig_B{y, n}};
    receive Y, B, sig_Y{y, n};
]
```

$$\mathrm{WCR} \vdash [\ \mathrm{InitWCR}(A, B)\ ]_A \ \Diamond \mathrm{Verify}(A, \mathrm{sig}_B\{m, n\})$$

2: Properties of Signatures

InitWCR(A, X) = [
    new m;
    send A, X, {m};
    receive X, A, {x, sig_X{m, x}};
    send A, X, sig_A{m, x}};
]

RespWCR(B) = [
    receive Y, B, {y};
    new n;
    send B, Y, {n, sig_B{y, n}};
    receive Y, B, sig_Y{y, n}};
]

$$\text{WCR} \vdash [\ \text{InitWCR}(A, B)\ ]_A \ \text{Honest}(B) \supset \exists t' (\Diamond \text{Send}(B, t') \wedge \text{Contains}(t', sig_B\{m, n\}))$$

Honesty Invariant

InitWCR(A, X) = [
 new m;
 send A, X, {m};
 receive X, A, {x, sig_X\{m, x\}};
 send A, X, sig_A\{m, x\};
]

RespWCR(B) = [
 receive Y, B, {y};
 new n;
 send B, Y, {n, sig_B\{y, n\}};
 receive Y, B, sig_Y\{y, n\};
]

$$\begin{aligned}\text{WCR} \mid\!- \;\; & \text{Honest}(X) \wedge \\ & \Diamond \text{Send}(X, t') \wedge \text{Contains}(t', \text{sig}_X\{y, x\}) \wedge \\ & \neg \Diamond \text{New}(X, y) \supset \\ & \Diamond \text{Receive}(X, \{Y, X, \{y\}\})\end{aligned}$$

In this protocol, sig_x\{y,x\} does not explicitly include identity of intended recipient Y

3: Use Honesty Rule

```
InitWCR(A, X) = [
    new m;
    send A, X, {m};
    receive X, A, {x, sig_X{m, x}};
    send A, X, sig_A{m, x};
]
```

```
RespWCR(B) = [
    receive Y, B, {y};
    new n;
    send B, Y, {n, sig_B{y, n}};
    receive Y, B, sig_Y{y, n};
]
```

$$\text{WCR} \vdash [\ \text{InitWCR}(A, B)\ ]_A\ \text{Honest}(B) \supset \Diamond\text{Receive}(B, \{Y, B, sig_Y\{y, n\}\})$$

B receives 3[rd] message from someone, not necessarily A

Failed Proof and Counterexample

- WCR does not provide the strong authentication property for the initiator
- Counterexample: intruder can forge sender's and receiver's identity in first two messages
  - A -> X(B) A, B, m
  - X(C) -> B C, B, m [X pretends to be C]
  - B -> X(C) n, $sig_B(m, n)$
  - X(B) -> A n, $sig_B(m, n)$

Further Work on Protocol Logic

- See papers by Datta, Derek, Mitchell, and Pavlovic on the course website
  - With a Diffie-Hellman primitive, prove authentication and secrecy for key exchange (STS, ISO-97898-3)
  - With symmetric encryption and hashing, prove authentication for ISO-9798-2, SKID3
- Work on protocol derivation
  - Build protocols by combining standard parts
    - Similar to the derivation of JFK described in class
  - Reuse proofs of correctness for building blocks
    - Compositionality pays off!