Protocol composition logic
symbolic model, computational model, and applications

John C Mitchell
Stanford University

Second lecture on PCL will be given by Anupam Datta

Stanford Security Lab

Protocol Composition Logic (PCL)

Security Lab

Summary

PCL is a logic for proving security properties of network protocols. Two central results for PCL are a set of composition theorems and a computational soundness theorem. In contrast to traditional folk wisdom in computer security, the composition theorems allow proofs of complex protocols to be built up from proofs of their constituent sub-protocols. The computational soundness theorem guarantees that, for a class of security properties and protocols, axiomatic proofs in PCL carry the same meaning as reduction-style cryptographic proofs. Tool implementation efforts are also underway. PCL and a complementary model-checking method have been successfully applied to a number of internet, wireless and mobile network security protocols developed by the IEEE and IETF Working Groups. This work identified serious security vulnerabilities in the IEEE 802.11i wireless security standard and the IETF GDOI standard. The suggested fixes have been adopted by the respective standards bodies.

PCL has been the topic of invited talks at premier venues including ASL'01, MFPS'03, ICALP'05, LCC'06, and ASIAN'06. It has been taught in security courses at a number of universities including Aachen, CMU, Penn, Stanford, and Texas. Three papers on this work have been invited to special issues of journals, which are compilations of the best papers presented at the respective venues.

The following paper and set of slides provides an overview of this project. For further details, please read the other papers included below.

- A. Datta, A. Derek, J. C. Mitchell, A. Roy, Protocol Composition Logic (PCL), *Electronic Notes in Theoretical Computer Science*, Volume 172 , 1 April 2007, Pages 311-358. Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin. [Paper]

- J. C. Mitchell, Symbolic and Computational Analysis of Network Protocol Security, *Invited Talk, ASIAN Computing Science Conference*, December 2006. [Slides]

Also see the model-checking page for related projects.

http://crypto.stanford.edu/protocols/

Goals

- PCL is an evolving research framework for investigating this basic question:

  Is it possible to prove security properties of current practical protocols using compositional, direct reasoning that does not mention the actions of the attacker?

- Direct reasoning

  If Alice creates and sends a nonce n and later receives Bob's signature on ⟨n,m⟩, then Bob signed ⟨n,m⟩ *after* Alice created and sent the nonce.

Goals

- Combine the advantages of BAN
  - Annotate programs with assertions
  - High-level direct reasoning
  - No explicit reasoning about attacker
- With accepted protocol semantics
  - Set of roles executed concurrently by principals
  - Attacker controls the network
  - *Eventually:* symbolic and computational semantics

Goals

- Case studies of IETF, IEEE standard protocols
  - *Eventually:* proofs of some kind for all major widely used network security protocols
  - SSL/TLS, WPA2, IKEv2, Kerberos (PK-init, ...), ...
- Even if
  - Some of these protocols only have “weak” security guarantees under “reasonable” assumptions about the crypto primitives they use

Non-Goal

- Full formal proofs in this decade
  - PCL has axioms, proof rules about protocol steps

$$[\ \textit{new}\ n\,]_P\ \text{“}\textit{Knows}(P,n)\text{”}$$

  - Includes a Rule of Consequence [Hoare...]

$$\frac{\varphi\ [\ \textit{actions}\ ]_P\ \psi \qquad \psi \supset \theta}{\varphi\ [\ \textit{actions}\ ]_P\ \theta}$$

but does not contain specific rules for $\psi \supset \theta$

Someone can do this later if everything else works out

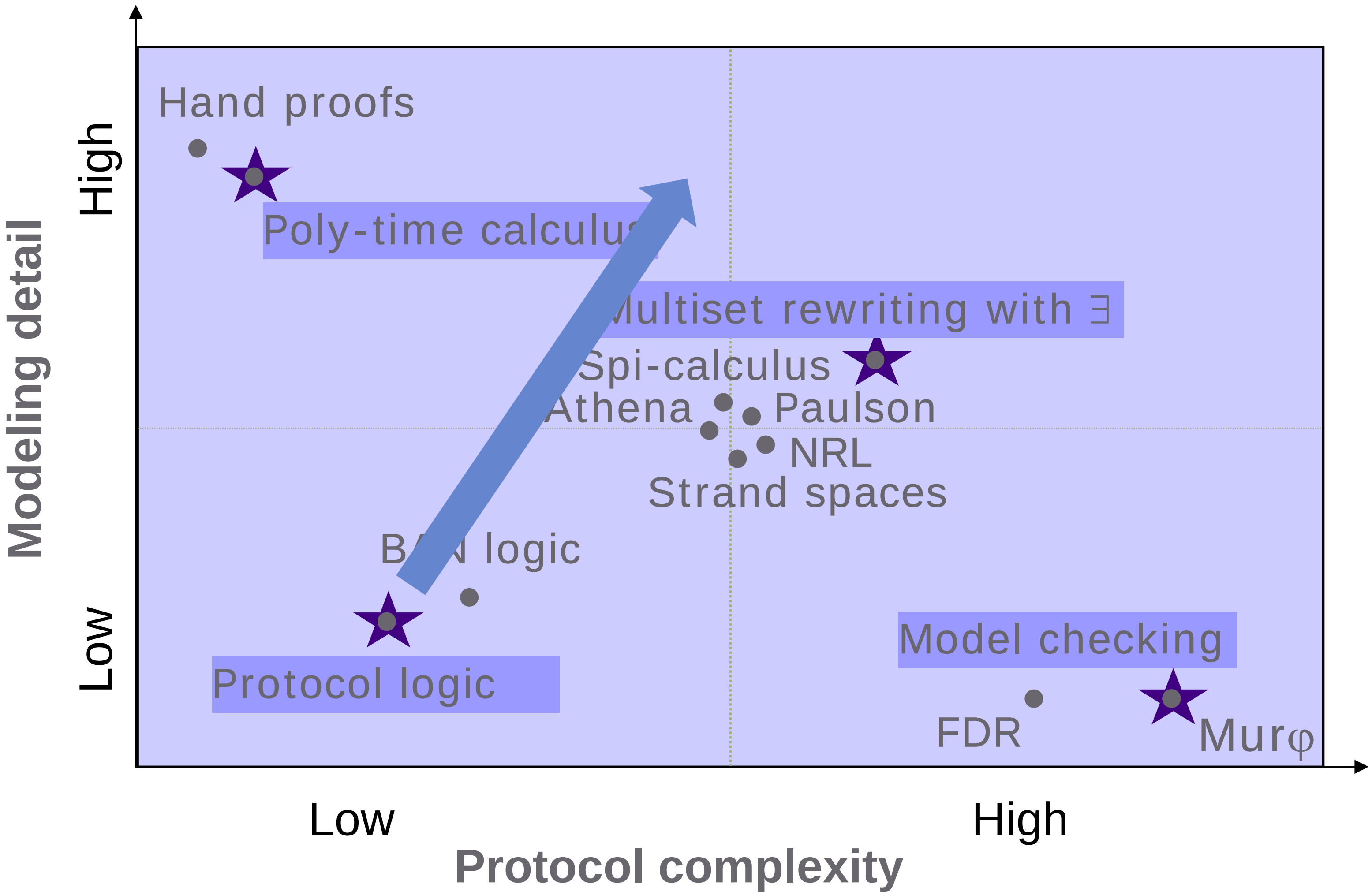
PCL has been a team effort

- Collaborators

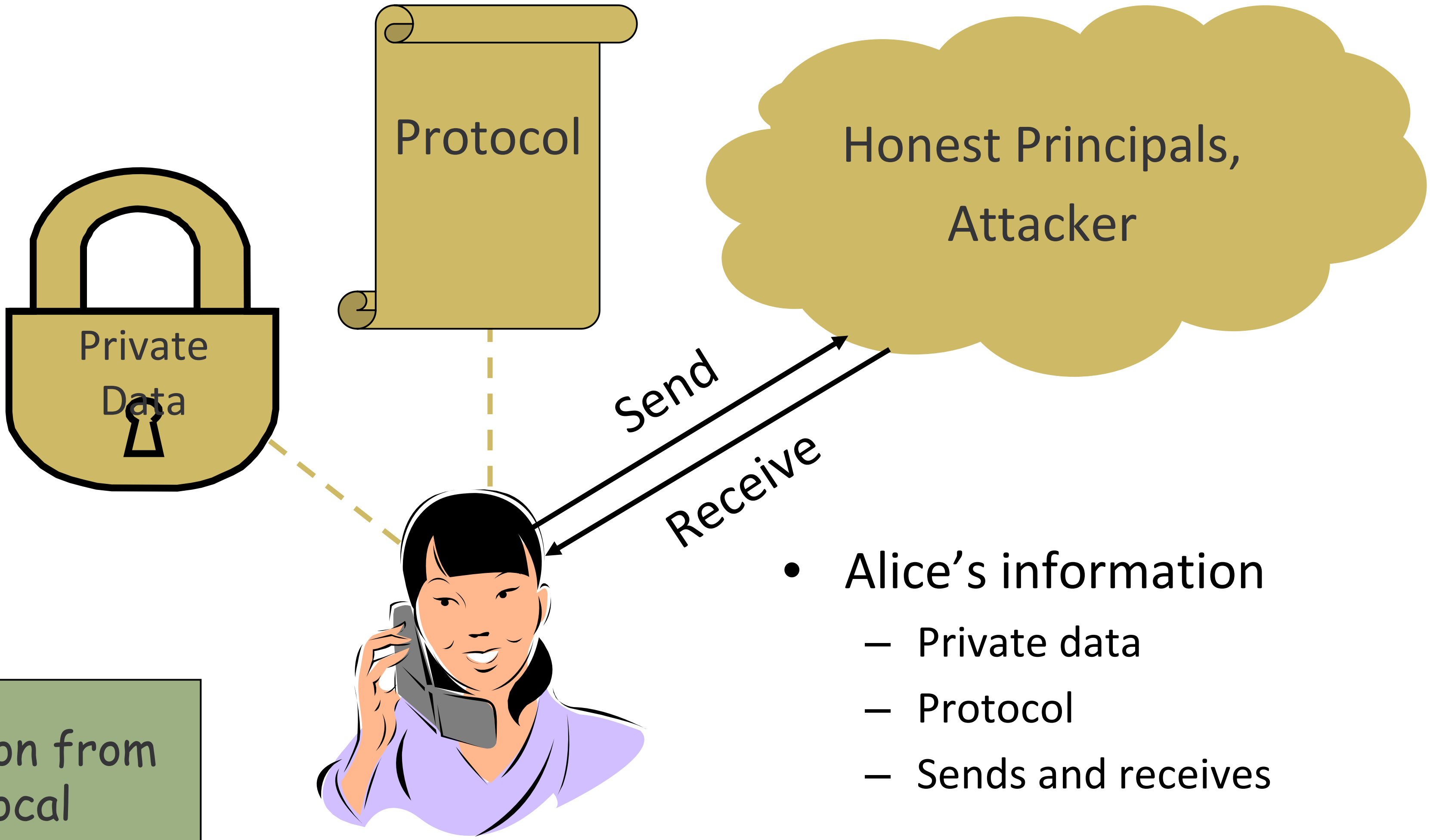
  M. Backes, A. Datta, A. Derek, N. Durgin, C. He,
  R. Kuesters, D. Pavlovic, A. Ramanathan, A. Roy,
  A. Scedrov, V. Shmatikov, M. Sundararajan, V. Teague,
  M. Turuani, B. Warinschi, …

Science is a social process

Protocol analysis spectrum



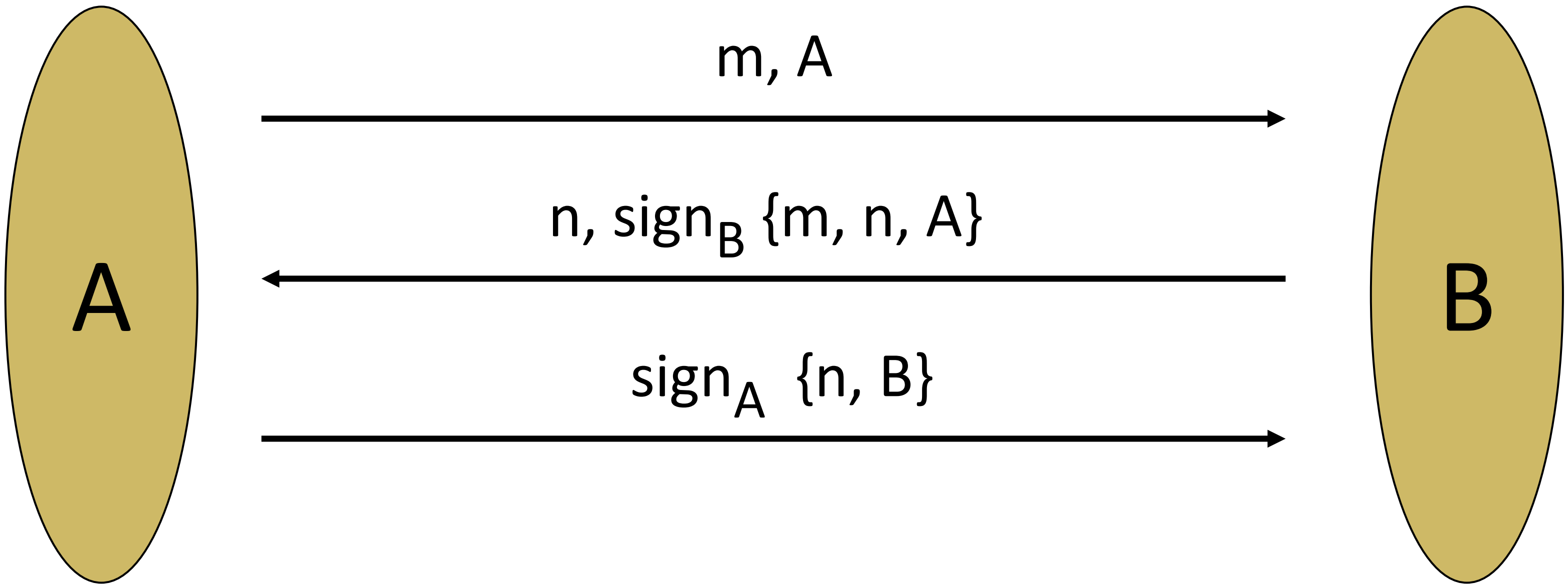
Protocol composition logic



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

Reason from local information

Challenge-Response Protocol



Running example for a number of slides.

Protocol logic: Actions

send m;	send message m
receive x;	receive a message into variable x
new n;	generate new nonce n

- A program is a sequence of actions

InitCR(A, B) = [
new m;
send A, B, ⟨m, A⟩;
receive B, A, n, sig_B{“r”, m, n, A}⟩;
send A, B, sig_A{“i”, m, n, B};
$]_A$

RespCR(B) = [
receive A, B, ⟨m, A⟩;
new n;
send B, A, ⟨n, sig_B{“r”, m, n, A}⟩;
receive A, B, sig_A{“i”, m, n, B};
$]_B$

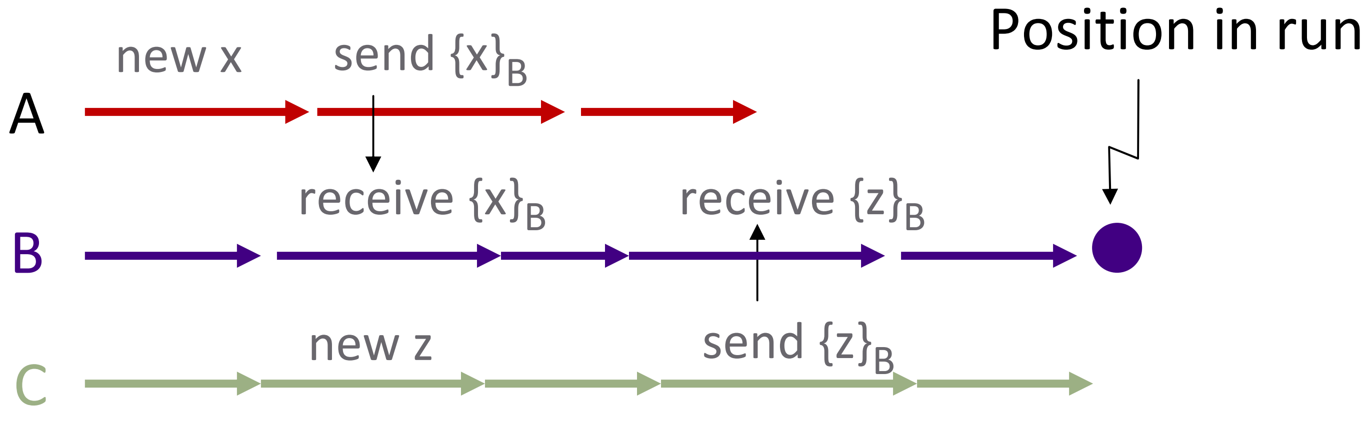
Example send action is send m’ where m’ = ⟨A, B, ⟨m, A⟩⟩ includes source and destination

Symbolic Attacker

- Controls complete network
  - Can read, remove, inject messages
- Fixed set of operations on terms
  - Pairing, Projection
  - Encryption with known key
  - Decryption with known key
  - …
- Represent attacker
  - by large set of attacker programs that can do these actions (c.f. “penetrator strands”)

Execution Model

- Initial Configuration
  – Set of principals and keys assigned to them
  – Assignment of ≥ 1 role to each principal
- Run
  – Interleaving of actions of honest principals and attacker, starting from initial configuration



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*] $_P\ \psi$

- Example

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

Specifies secrecy

Challenge-Response Property

Specifying authentication for Responder

CR $|=$ true [RespCR(B) $]_B$ Honest(A) $\supset$ (

Send(A, ⟨A,B, ⟨m, A⟩⟩) < Receive(B, ⟨A,B, ⟨m, A⟩⟩) $\wedge$

Receive(B, ⟨A,B,⟨m, A⟩⟩) < Send(B, ⟨B,A, ⟨n, sig_B {"r",m, n, A}⟩⟩) $\wedge$

Send(B, ⟨B,A, ⟨n, sig_B {"r",m, n, A}⟩⟩) < Receive(A, ⟨ B,A, ⟨n, sig_B {"r",m, n, A}⟩⟩) $\wedge$

Receive(A, ⟨ B,A, ⟨n, sig_B {"r",m, n, A} ⟩⟩) < Send(A, ⟨ A,B,{sig_A{"i",m,n,B} ⟩⟩) $\wedge$

Send(A, ⟨ A,B, sig_A{"i",m,n,B} ⟩) < Receive(B, ⟨ A,B, sig_A{"i",m,n,B} ⟩)

)

"Actions in Order"

Authentication as "matching conversations" [Bellare-Rogaway93]

Proof System

- Prove security properties of useful protocols
- Axioms
  - Simple formulas about actions, etc.
- Inference rules
  - Proof steps
- Theorem
  - Formula obtained from axioms by application of inference rules

Core concept: Honesty

- A *principal* X is *honest* in run R if
  - Intuitively,

    "X only does what X is supposed to do"
  - More precisely

    The actions of X in R are precisely an interleaving of initial segments of traces of a set of roles of the protocol

We assume that protocols do not reveal pre-assigned keys of any principal.
Certain axioms and rules are sound only under this assumption.
These axioms and rules can be dropped and replaced if the assumption is dropped.

Sample axioms

- Actions

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

- Public key encryption

  Honest(X) $\wedge$ Decrypt(Y, enc_X\{m\}) $\supset$ X=Y

- Signature

  Honest(X) $\wedge$ Verify(Y, sig_X\{m\})

  $\supset$ Sign(X, sig_X\{m\})

Authentication for CR Responder – part 1

InitCR(A, B) = [

new m;

send A, B, ⟨m, A⟩;

receive B, A, ⟨n, sig_B{"r", m, n, A}⟩;

send A, B, sig_A{"i", m, n, B};

$]_A$

RespCR(B) = [

receive A, B, ⟨m, A⟩;

new n;

send B, A, ⟨n, sig_B{"r", m, n, A}⟩;

receive A, B, sig_A{"i", m, 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) $]_B$ Honest(A) ⊃ Sign(A, sig_A{"i", m, n, A})

Proving Invariants

- We want to prove

  $\Gamma \equiv$ Honest(A) $\supset \varphi$,

  where $\varphi \equiv$

  (Sign(A, sig_A("i", m, n, B)) $\rightarrow$ Receive(A, $\langle$n, sig_B("r", m, n, A) $\rangle$))

- "φ holds at all pausing states of all honest roles"
  - protocol segment: subsequence of honest party actions between pausing states
  - Picture of when invariant φ holds:

    φ --- actions of A --- φ ---- actions of B ---
    φ --- attacker actions --- φ ---- actions of B --- φ --- …

Why is this an invariant of CR?

InitCR(A, B) = [
    new m;
    send A, B, ⟨m, A⟩;
    receive B, A, ⟨n, sig_B{“r”, m, n, A}⟩;
    send A, B, sig_A{“i”, m, n, B};
$]_A$

RespCR(B) = [
    receive A, B, ⟨m, A⟩;
    new n;
    send B, A, ⟨n, sig_B{“r”, m, n, A}⟩;
    receive A, B, sig_A{“i”, m, n, B};
$]_B$

- Honest behavior
  - One or more instances of these two roles
- Property of each role
  - If A signs sig_A(“i”, m, n, B))

    A must be executing InitCR role

    A previously received ⟨B, A ⟨n, sig_X{“r”, m, n, A}⟩⟩;

Honesty Rule

- Rule for establishing invariants:
  - Prove φ holds when threads are started
  - Prove, for all *protocol segments*, if φ held at the beginning, it holds at the end

InitCR(A, B) = [

seg 1:
new m;
send A, B, ⟨m, A⟩;

seg 2:
receive B, A, ⟨n, sig_B{"r", m, n, A}⟩;
send A, A, sig_A{"i", m, n, B);

$]_A$

RespCR(B) = [

seg 3:
receive A, B, ⟨m, A⟩;
new n;
send B, A, ⟨n, sig_B{"r", m, n, A}⟩;

seg 4:
receive A, B, sig_A{"i", m, n, B};

$]_B$

We have formulated more than one honesty rule, plus secrecy induction.
Eventually: we would like to unify these rules.

Authentication for CR Responder – part 2

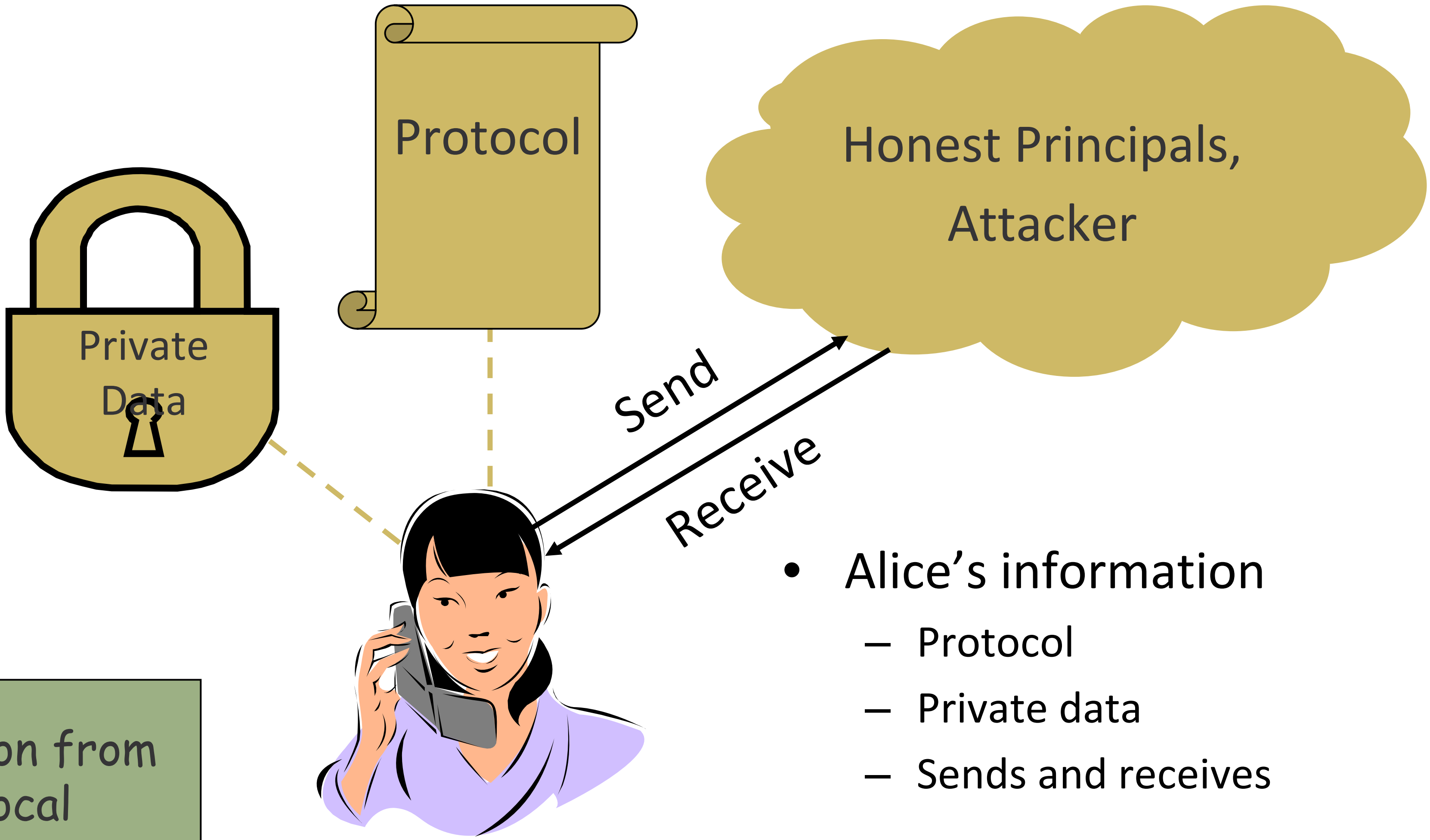
- So far
  - CR |- true [RespCR(B) $]_B$ Sign(A, sig_A{“i”, m, n, A})
- Use invariant Γ to prove:
  - CR |- true [RespCR(B) $]_B$ Receive(A, n, sigB{“r”, m, n, A})
- Reason from B’s point of view to prove:
  - CR |- true [RespCR(B) $]_B$ FirstSend(B, n, $\langle$n, sigB{“r”, m, n, A}$\rangle$))
- Apply Nonce freshness axiom to prove:
  - CR |- true [RespCR(B) $]_B$

    Receive(A, $\langle$n, sigB{“r”, m, n, A} $\rangle$) < Send(B, sigB{“r”, m, n, A})
- Additional similar steps complete the proof

Sample PCL studies

- Wireless 802.11i
  - Model checking to find errors, improve
  - PCL proof of correctness, including TLS
- Kerberos
  - Including variants “PK-Init” and “DH-init”
- Extensible Authentication Protocol (EAP)
  - Model check to find errors, improve
  - PCL proof of correctness, identify subtleties
- Mesh Security Architecture (IEEE 802.11s)
  - *Motorola group* added some axioms, found problems, identified invariants, proved correctness

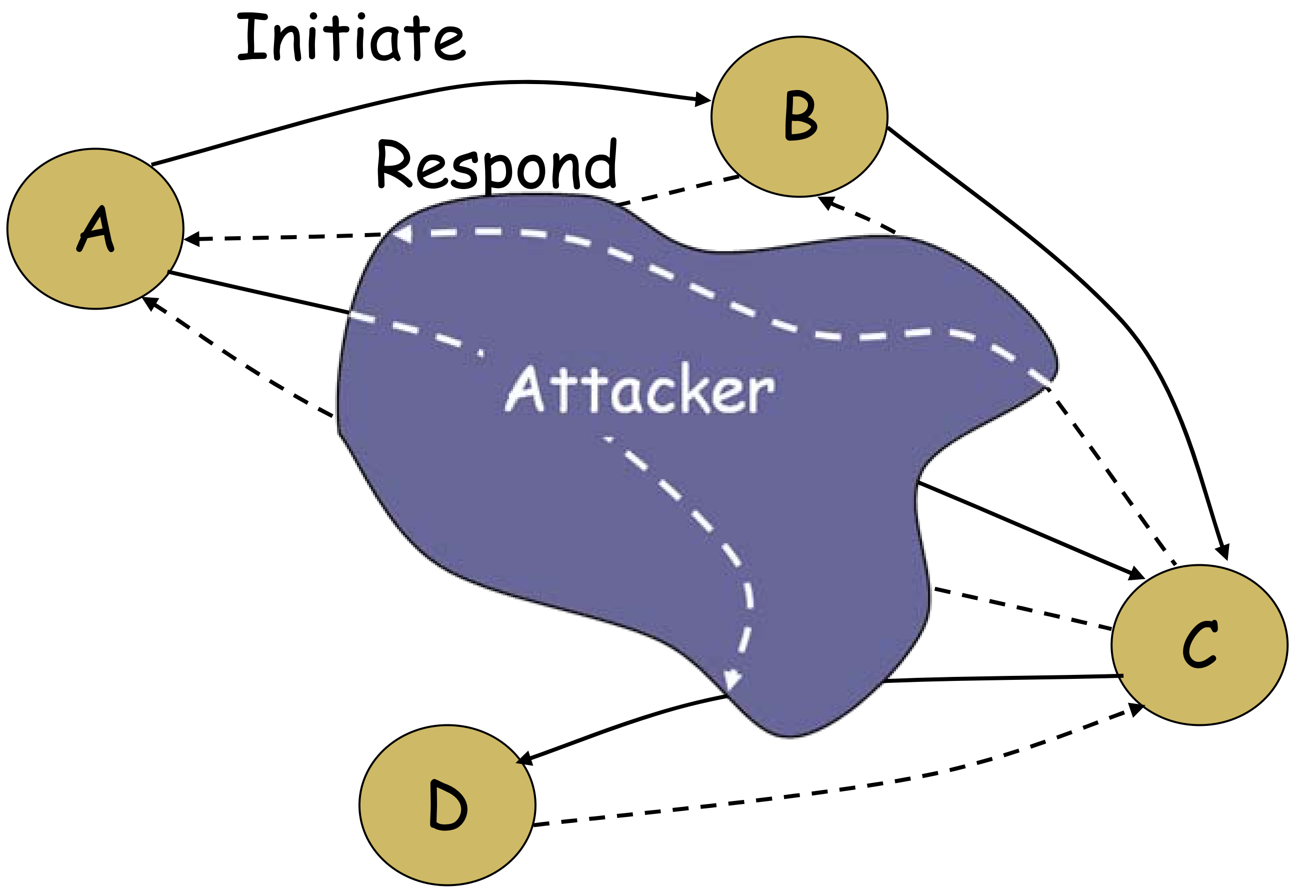
Protocol composition logic



- Alice’s information
  - Protocol
  - Private data
  - Sends and receives

Reason from local information

Principal may execute many roles



Some Details

- Protocol
  - Given by a set of roles
- Role
  - Program for one participant in protocol
  - Example:
    - InitCR(A,B) : A initiates Challenge-Response with B
- Principal
  - Agent, associated with a key pair, signing key, and/or symmetric key
- Thread
  - A role, instantiated and executed by a principal
  - Semantically: Principal, role instance, unique thread ID

Some Details

- Notation in PCL papers
  - Threads X, Y, Z, ... Executed by principals $\hat{X}$, $\hat{Y}$, $\hat{Z}$, ...
  - Some abuse of notation for readability (order-sorted algebra) ...

$$\text{InitCR}(A,X) = [\ ....\ \text{sig}_A\{\text{“i”}, m, n, B\}\ ...\ ]_A$$

Principal, key associated with principal, thread

Formulas

- Action Formulas

  a ::= Send(X,m) | Receive(X,m) | ...

  principal X sends message m in thread X

- Formulas

  φ ::= a | a < a' | Has(X,m) | Fresh(X,m) | Honest(N) | ...

  an action a happens before an action a'

  principal N is honest

- Modal formulas

  Ψ ::= φ [actions $]_X$ ψ

  if φ before, then after thread X completes actions, ψ

Semantics

- Protocol Q
  - Provides set of roles (e.g., initiator, responder)
- Run R of Q
  - 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

Formula φ satisfied by protocol Q at run R

- Defined by induction on formula φ

  $Q,R \models Send(X,m)$ if thread X sent m in R ...

  $Q,R \models Honest(\hat{X})$ if $\hat{X}$ is an honest principal in the initial configuration of R and $R|_{\hat{X}}$ is an interleaving of basic sequences of instances of roles of Q

  ...

  $Q,R \models \varphi\ [\ P\ ]_X\ \psi$ if for all partitions $R = R_0R_1R_2$ and all substitutions σ, if $Q,R_0 \models \sigma\varphi$ and σ' matches P to $R_1|_X$ then $Q,R_2 \models (\sigma \bullet \sigma')\psi$

The first substitution is a symbolic environment giving values to variables.
The second accounts for how P uses variables and the way operations in P bind variabes in ψ.

Core concept: $[\ \ldots\ ]_X$

$[\ a_1\ a_2\ a_3\ \ldots\ ]_X\ \psi$ vs $a_1^X < a_2^X < a_3^X < \ldots \supset \psi$

where if a_i = send m then a_i^X = Send(X,m)

- Modal form
  - Thread X did a_1 a_2 a_3 ... in this order, with no other actions interleaved
- Non-modal form
  - Thread X did a_1 a_2 a_3 ... in order, but might have done some other things too in between or after

Proving absence of actions

- Some axioms

  $\text{Start } []_X \neg a^X$

  $\neg a^X \, [b]_X \neg a^X$ provided a, b do not unify

- Relevant proof rule

$$\frac{\varphi \, [\, S \,]_X \, \psi \qquad \psi \;\, [\, T \,]_X \, \theta}{\varphi \, [\, ST \,]_X \, \theta}$$

$$\frac{\text{Start } [a_1 \, a_2]_X \neg b^X \qquad \neg b^X [a_3 \,]_X \neg b^X}{\text{Start } [a_1 \, a_2 \, a_3 \,]_X \neg b^X}$$

- In contrast,

  $a_1{}^X < a_2{}^X < a_3{}^X \supset \neg \, b^X$ is invalid

Honesty rule (rule scheme)

$$\frac{\text{Start}(X)\ [\ ]_X\ \phi \qquad \forall \text{roles } R \text{ of } Q.\ \forall \text{ protocol segments } S \text{ of } R.\ \ \phi\ [\ S\ ]_X\ \phi}{Q \mid\!- \text{Honest}(X) \supset \phi}$$

This is a finitary rule:

- Typical protocol has 2-3 roles
- Typical role has 1-3 receives
- Only need to consider A waiting to receive

Honesty rule (example use)

$$\frac{\text{Start}(X)\ [\ ]_X\ \phi \qquad \begin{array}{l}\forall \text{roles } R \text{ of } Q.\\ \forall \text{ protocol segments } S \text{ of } R.\\ \phi\ [\ S\ ]_X\ \phi\end{array}}{Q \mid\!- \text{Honest}(X) \supset \phi}$$

How this can be used

- If Y receives a message m from X, and

  $\text{Honest}(X) \supset (\text{Sent}(X,m) \supset \text{Received}(X,m'))$

- Then Y can conclude

  $\text{Honest}(X) \supset \text{Received}(X,m')$

Principal Y can draw conclusions about another principal, X.

Example: Honesty Rule for CR

InitCR(A, B) = [

seg 1:
- new m;
- send A, B, ⟨m, A⟩;

seg 2:
- receive B, A, ⟨n, sig_B{"r", m, n, A}⟩;
- send A, B, sig_A{"i", m, n, B}; (msg 3)

$]_A$

RespCR(B) = [

seg 3:
- receive A, B, ⟨m, A⟩;
- new n;
- send B, A, ⟨n, sig_B{"r", m, n, A}⟩;

seg 4:
- receive A, B, sig_A{"i", m, n, B};

$]_B$

For seg 2:

Sent(X,m3) ⊃ Received(X,m2)

[receive X, A, ⟨x, sig_X{"r", m, x, A}⟩;$]_X$

Received(X,m2)

Sent(X,m3) ⊃ Received(X,m2)

[receive X, A, ⟨x, sig_X{"r", m, x, A}⟩; send A, X, sig_A{"i", m, x, X};$]_X$

Received(X,m2

Received(X,m2) ⊃ (Sent(X,m3) ⊃ Received(X,m2))

For other segments, prove ¬(Sent(X,m3) and derive **(Sent(X,m3) ⊃ Received(X,m2))**

Example complete PCL proof for InitCR

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	$[\langle A, B, m\rangle]_{A,\eta}\ \Diamond \mathsf{Send}(A, \{A, B, m\}, \eta)$
AA1	$[(B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}})]_{A,\eta}$
	$\Diamond \mathsf{Receive}(A, \{B, A, n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}, \eta)$
AA1	$[(\{\!\|m, n, A\|\!\}_{\overline{B}} / \{\!\|m, n, A\|\!\}_{B})]_{A,\eta}\ \Diamond \mathsf{Verify}(A, \{\!\|m, n, A\|\!\}_{\overline{B}}, \eta)$
AA1	$[\langle A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\rangle]_{A,\eta}\ \Diamond \mathsf{Send}(A, \{A, B, \{\!\|m, n, B\|\!\}_{\overline{A}}\}, \eta)$
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	$\Diamond \mathsf{New}(A, m, \eta) \supset \neg \Diamond \mathsf{New}(B, m, \eta')$
5, **VER**	$\mathsf{Honest}(B) \wedge \Diamond \mathsf{Verify}(A, \{\!\|m, n, A\|\!\}_{\overline{B}}, \eta) \supset$
	$\exists \eta'.\exists m'.(\Diamond \mathsf{CSend}(B, m', \eta') \wedge (\{\!\|m, n, A\|\!\}_{\overline{B}} \subseteq m'))$
HON	$\mathsf{Honest}(B) \supset (\exists \eta'.\exists m'.((\Diamond \mathsf{CSend}(B, m', \eta') \wedge$
	$\{\!\|m, n, A\|\!\}_{\overline{B}} \subseteq m' \wedge \neg \Diamond \mathsf{New}(B, m, \eta')) \supset$
	$(\ m' = \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\} \wedge \Diamond \mathsf{Receive}(B, \{A, B, m\}, \eta') \wedge$
	$\mathsf{ActionsInOrder}(\mathsf{Receive}(B, \{A, B, m\}, \eta'), \mathsf{New}(B, n, \eta'),$
	$\mathsf{Send}(B, \{B, A, \{n, \{\!\|m, n, A\|\!\}_{\overline{B}}\}\}, \eta')))))$
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

We have a PCL proof. So what?

- Soundness Theorem:
  - If Q |- ϕ then Q |= ϕ
  - If ϕ is a theorem of PCL then ϕ is a valid formula

- Valid: ϕ holds in any step in any run of protocol Q
  - Unbounded number of participants
  - Dolev-Yao intruder
  - Possibly also for computational model (CPCL)

Using PCL for simple protocols: summary

- Model the protocol
  - Program for each protocol role
- Express security properties
  - Using PCL syntax
  - Authentication, secrecy easily expressed
- Prove security properties
  - Using PCL proof system
    - Using sound implications of pre-conditions and post-conditions
  - Soundness theorem guarantees that provable properties hold in all protocol runs

Protocol composition

- Sequential composition of protocols
  - Run key-exchange protocol
  - Then protocol that uses keys
- Parallel composition
  - Run two protocols in parallel
    - $Q_1 \mid Q_2$: union of the sets of roles of Q_1 and Q_2
  - Examples:
    - Many protocols run in parallel, e.g., SSL, IKE, Kerberos
    - In 802.11i, TLS, 4WAY, GroupKey can be run in parallel

Sequential Composition

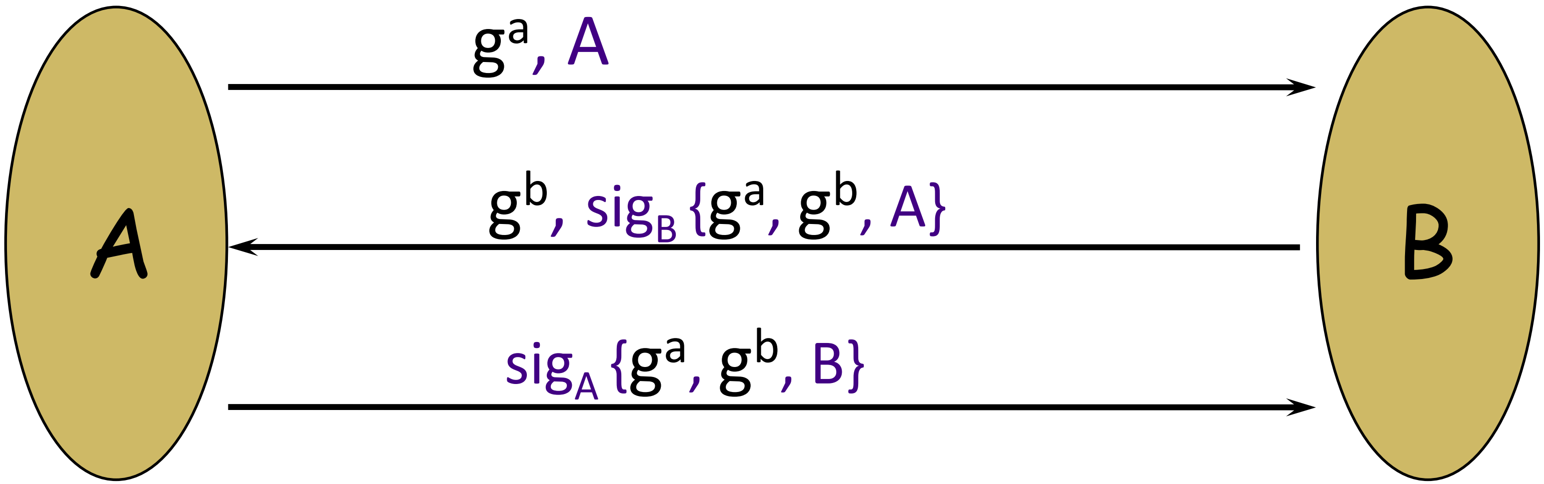
- Composition rule

$$\frac{\varphi\ [\ S\ ]_P\ \psi \qquad \psi\ [\ T\ ]_P\ \theta}{\varphi\ [\ ST\ ]_P\ \theta}$$

- What else do we need?
  - This rule lets us combine local reasoning about sequences of actions
  - But Honesty Rule (invariants) depend on entire protocol
  - How can we combine proofs of invariants?

Same problems for parallel composition

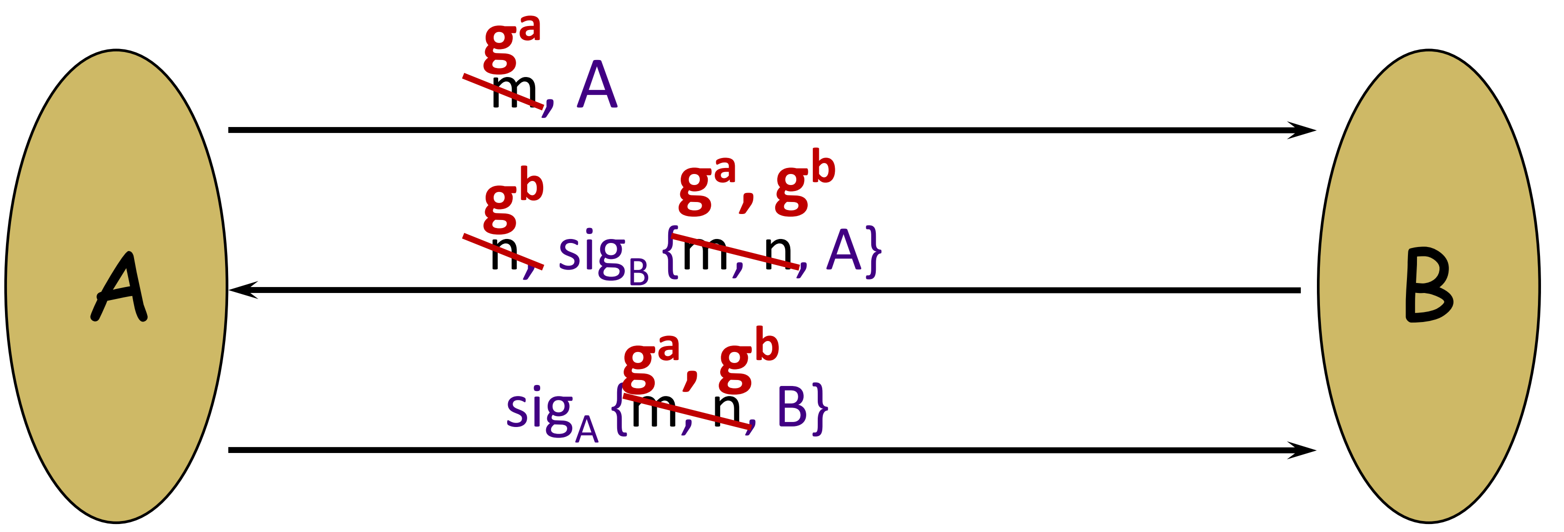
Example: ISO-9798-3



- Shared secret: g^{ab}
- Authentication
  - Similar to challenge-response
  - Do we need to prove property from scratch?

ISO 9798-3

~~Challenge-Response~~



- Shared secret: g^{ab}
- Authentication
  - Similar to challenge-response
  - Do we need to prove property from scratch?

Sequential Composition

DH

X, Y

↓

new x

↓

X, Y, g^x, x

CR-Init

W, Z, w, x

↓

send W, Z, w, A;
receive Z, W, z, $sig_Y\{w, z, W\}$;
send W, Z, $sig_X\{w, z, Z\}$;

↓

X, Y, z^x

ISO

X, Y

↓

new x;

send X, Y, g^x, A;
receive Y, X, z, $sig_Y\{g^x, z, X\}$;
send X, Y, $sig_X\{g^x, z, Y\}$;

↓

X, Y, z^x

Sequential composition of roles with term substitution

Abstract challenge response

InitACR(A, X, m) = [
    send A, X, {m};
    receive X, A, ⟨x, sig_X{m, x}⟩;
    send A, X, sig_A{m, x}};
$]_A$

RespACR(B, n) = [
    receive Y, B, {y};
    send B, Y, ⟨n, sig_B{y, n}⟩;
    receive Y, B, sig_Y{y, n}};
$]_B$

- Role parameters m and n instead of nonces
- Specification by modal form: φ [actions] ψ
  - precondition: Fresh(A,m)
  - actions: $[\ \mathrm{InitACR}\ ]_A$
  - postcondition: Honest(B) $\supset$ Authentication
- Secrecy is proved from properties of Diffie-Hellman

Diffie-Hellman: Property

- Formula
  - true [new a]$_A$ Fresh(A, g^a)

- Diffie-Hellman property:
  - Can compute g^{ab} given g^a and b or g^b and a
  - Cannot compute g^{ab} given g^a and g^b

Composition: DH+CR = ISO-9798-3

- Additive Combination
  - DH post-condition matches CR precondition
  - Sequential Composition:
    - Substitute g^a for m in CR to obtain ISO.
    - Apply composition rule
    - ISO initiator role inherits CR authentication.
  - DH secrecy is also preserved
    - Proved using another application of composition rule.
- Nondestructive Combination
  - DH and CR satisfy each other's invariants

Parallel Composition Theorem (1)

- Honesty rule:

$$\frac{\forall \text{roles } R \text{ of } Q.\quad \forall \text{ protocol steps } A \text{ of } R.\quad \text{Start}(X)\ [\ ]_X\ \phi \qquad \phi\ [\ A\ ]_X\ \phi}{Q \mid\!- \text{Honest}(X) \supset \phi}$$

- Lemma:

  Let $Q = Q_1 \mid Q_2$. If $Q_1 \mid\!- \phi$ by proof ending in single use of honesty rule and $Q_2 \mid\!- \phi$ similarly, then $Q \mid\!- \phi$

- Proof idea:

  Roles (Q) = Roles (Q1) $\cup$ Roles(Q2)

Parallel Composition Theorem (2)

- Theorem:

  Let $Q = Q_1 \mid Q_2$. If $Q_1 \mid\!- \Gamma$, $\Gamma \mid\!- \Psi$ and $Q_2 \mid\!- \Gamma$, then $Q \mid\!- \Psi$, where Γ includes all invariants proved using Honesty rule

  – Proof idea:
    - By Lemma, $Q \mid\!- \Gamma$
    - Also, $\Gamma \mid\!- \Psi$
    - Intuitively, the old proof tree for Q_1 still works

General composition pattern

Γ Γ'

DH ▶ Honest(X) ⊃ …   CR ▶ Honest(X) ⊃ …

Γ |- Secrecy Γ' |- Authentication

$\Gamma \cup \Gamma'$ |- Secrecy $\Gamma \cup \Gamma'$ |- Authentication

$\Gamma \cup \Gamma'$ |- Secrecy $\wedge$ Authentication [additive]

DH • CR ▶ $\Gamma \cup \Gamma'$ [nondestructive]

=

ISO ▶ Secrecy $\wedge$ Authentication

Another composition pattern: Protocol Template

Challenge-Response Template

$A \rightarrow B$: m
$B \rightarrow A$: n, F(B,A,n,m)
$A \rightarrow B$: G(A,B,n,m)

Abstraction

ISO-9798-2

$A \rightarrow B$: m
$B \rightarrow A$: n, E_{KAB}(n,m,B)
$A \rightarrow B$: E_{KAB}(n,m)

SKID3

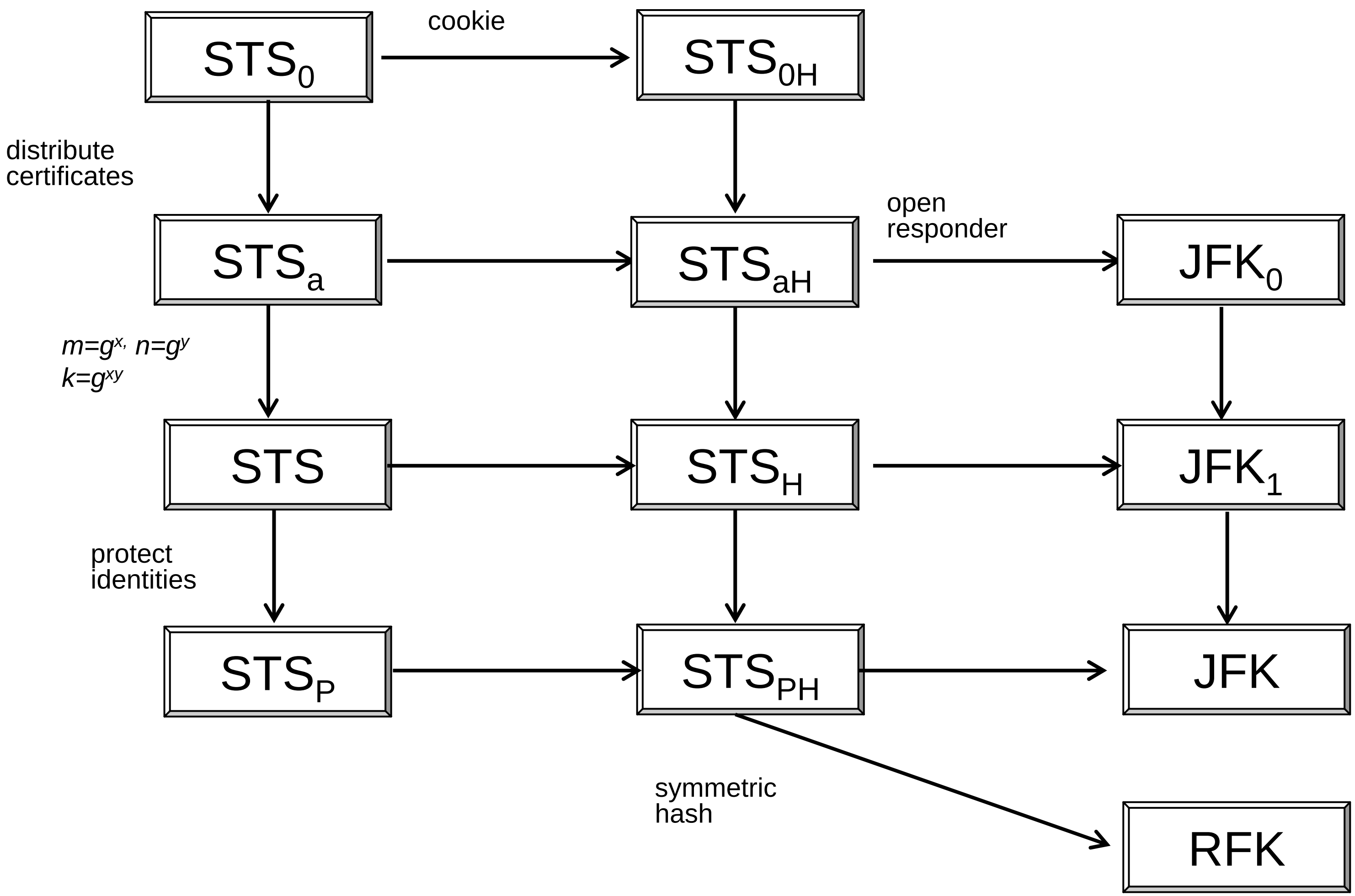
$A \rightarrow B$: m
$B \rightarrow A$: n, H_{KAB}(n,m,B)
$A \rightarrow B$: H_{KAB}(n,m,A)

ISO-9798-3

$A \rightarrow B$: m
$B \rightarrow A$: n, sig_B(n,m,A)
$A \rightarrow B$: sig_A(n,m,B)

Instantiation

STS family

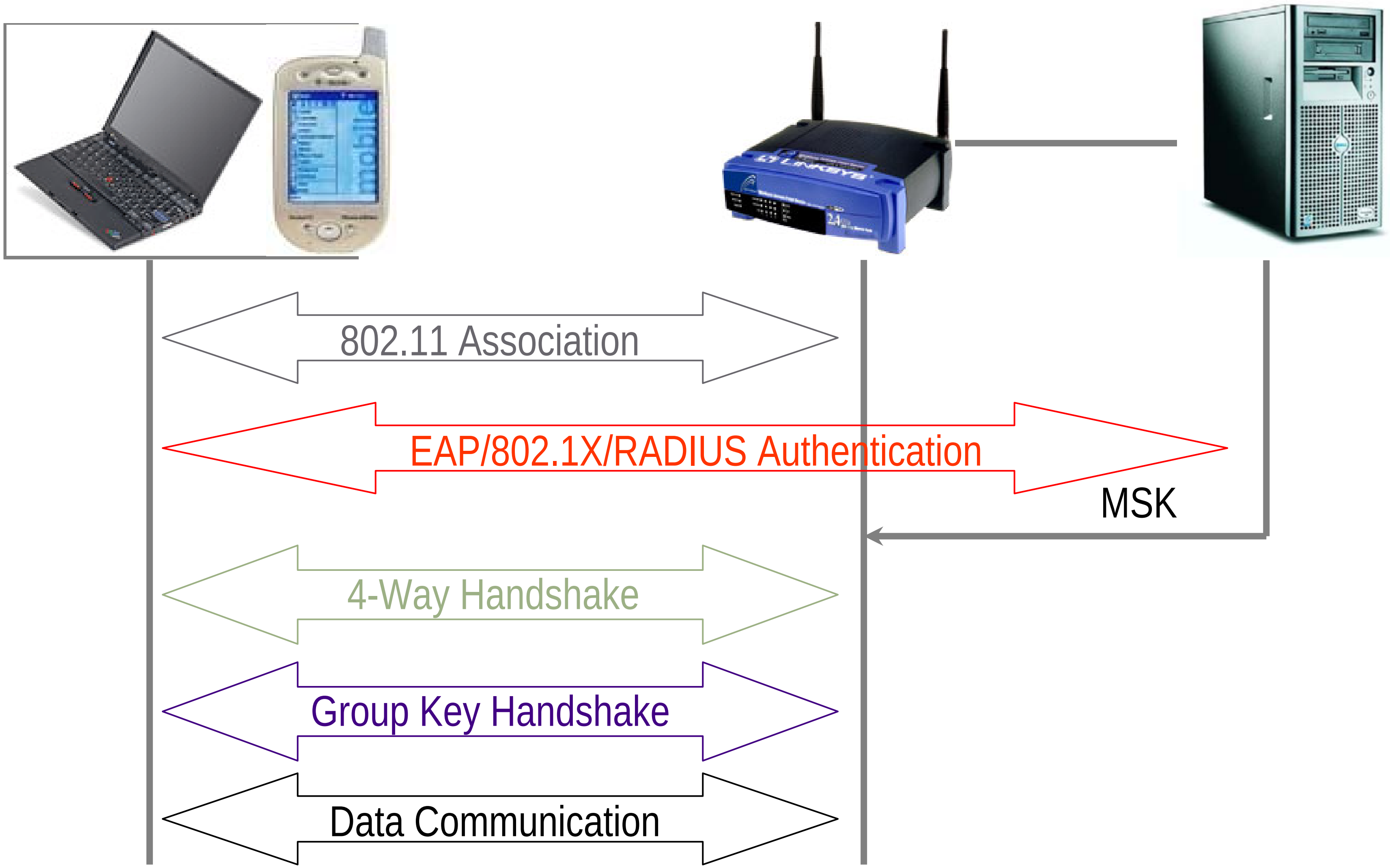


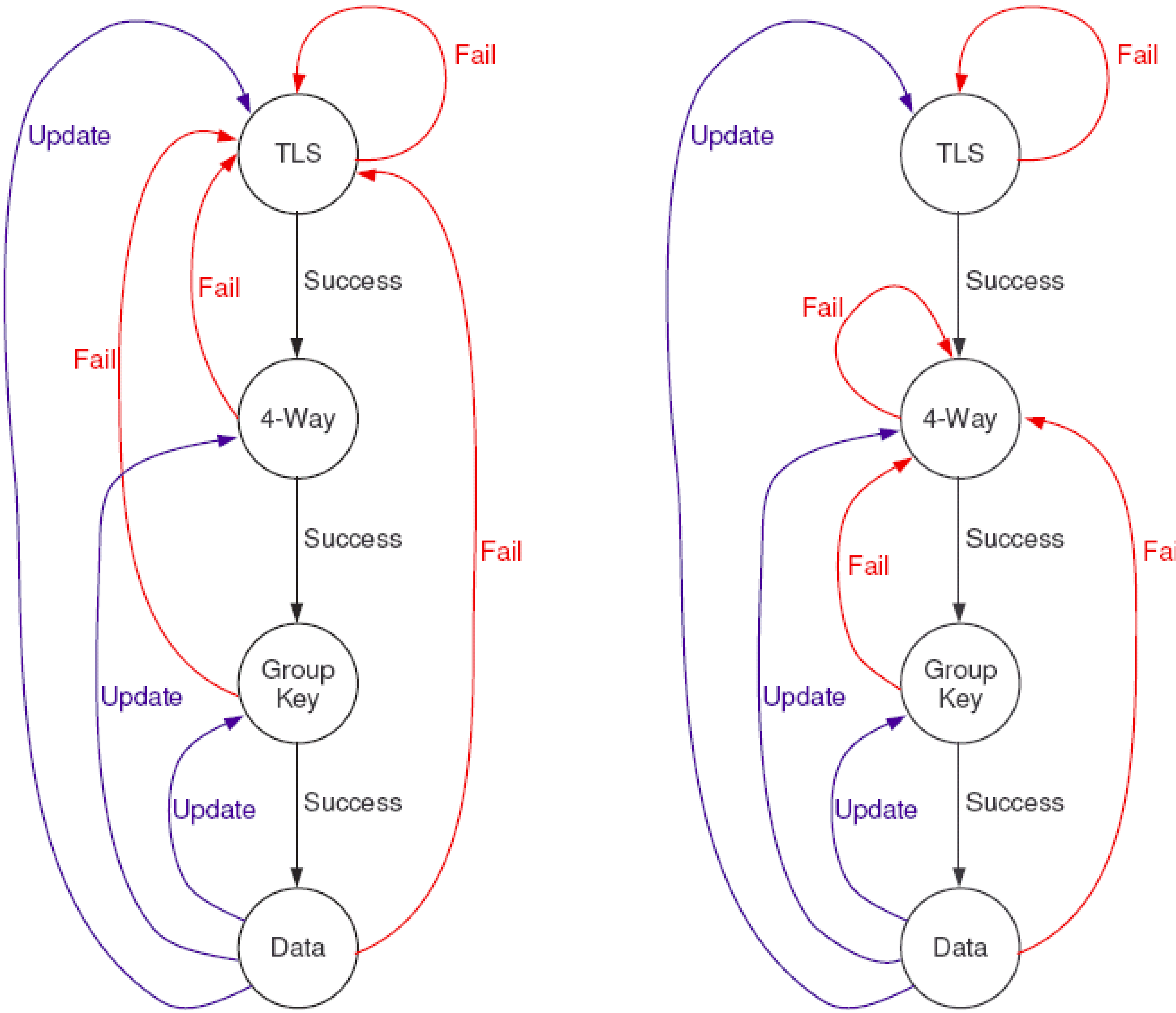
STS0
cookie
STS0H
distribute certificates
STSa
STSaH
open responder
JFK0
m=gx, n=gy
k=gxy
STS
STSH
JFK1
protect identities
STSP
STSPH
JFK
symmetric hash
RFK

Sample PCL studies

- Wireless 802.11i
  - Model checking to find errors, improve
  - PCL proof of correctness, including TLS
- Kerberos
  - Including variants “PK-Init” and “DH-init”
- Extensible Authentication Protocol (EAP)
  - Model check to find errors, improve
  - PCL proof of correctness, identify subtleties
- Mesh Security Architecture (IEEE 802.11s)
  - *Motorola group* added some axioms, found problems, identified invariants, proved correctness

802.11i Wireless Authentication





(a) Original Failure Recovery

(b) Improved Failure Recovery

Protocol Composition Logic: PCL

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