A Derivation System for Security Protocols and its Logical Formalization

Anupam Datta Ante Derek
John C. Mitchell Dusko Pavlovic

Stanford University Kestrel Institute

CSFW July 1, 2003

Contributions

- Protocol derivation
  - Build security protocols by combining parts from standard sub-protocols.
- Proof of correctness
  - Prove protocols correct using logic that follows steps of derivation.

Outline

- Derivation System
  - Motivating examples
  - Main concepts
  - Benefits
- Compositional Logic
  - Main idea
  - Syntax, semantics and proof system
  - Formalizing Composition
- Conclusions and Future Work

Protocol Derivation System

Example

- Construct protocol with properties:
  - Shared secret
  - Authenticated
  - Identity Protection
  - DoS Protection
- Design requirements for IKE, JFK, IKEv2 (IPSec key exchange protocol)

Component 1

- Diffie-Hellman

$$A \rightarrow B: \quad g^a$$

$$B \rightarrow A: \quad g^b$$

  - Shared secret (with someone)
    - A deduces:

$$\text{Knows}(Y, g^{ab)} \supset (Y = A) \vee \text{Knows}(Y,b)$$

  - Authenticated
  - Identity Protection
  - DoS Protection

Component 2

- Challenge Response:

$$A \rightarrow B: \; m, A$$
$$B \rightarrow A: \; n, sig_B\{m, n, A\}$$
$$A \rightarrow B: \; sig_A\{m, n, B\}$$

  - Shared secret (with someone)
  - Authenticated
    - A deduces: Received (B, msg1) Λ Sent (B, msg2)
  - Identity Protection
  - DoS Protection

Composition

$m := g^a$

$n := g^b$

- ISO 9798-3 protocol:

$A \rightarrow B$: g^a, A

$B \rightarrow A$: g^b, $sig_B\{g^a, g^b, A\}$

$A \rightarrow B$: $sig_A\{g^a, g^b, B\}$

  - Shared secret: g^{ab}
  - Authenticated
  - Identity Protection
  - DoS Protection

Refinement

- Encrypt signatures:

$$A \rightarrow B: \; g^a, A$$

$$B \rightarrow A: \; g^b, E_K\{sig_B\{g^a, g^b, A\}\}$$

$$A \rightarrow B: \; E_K\{sig_A\{g^a, g^b, B\}\}$$

  - Shared secret: g^{ab}
  - Authenticated
  - Identity Protection
  - DoS Protection

Transformation

- Use cookie: JFK core protocol

  $A \rightarrow B$: g^a, A

  $B \rightarrow A$: g^b, $hash_{KB}\{g^b, g^a\}$

  $A \rightarrow B$: g^a, g^b, $hash_{KB}\{g^b, g^a\}$

  $E_K\{sig_A\{g^a, g^b, B\}\}$

  $B \rightarrow A$: $g^b, E_K\{sig_B\{g^a, g^b, A\}\}$

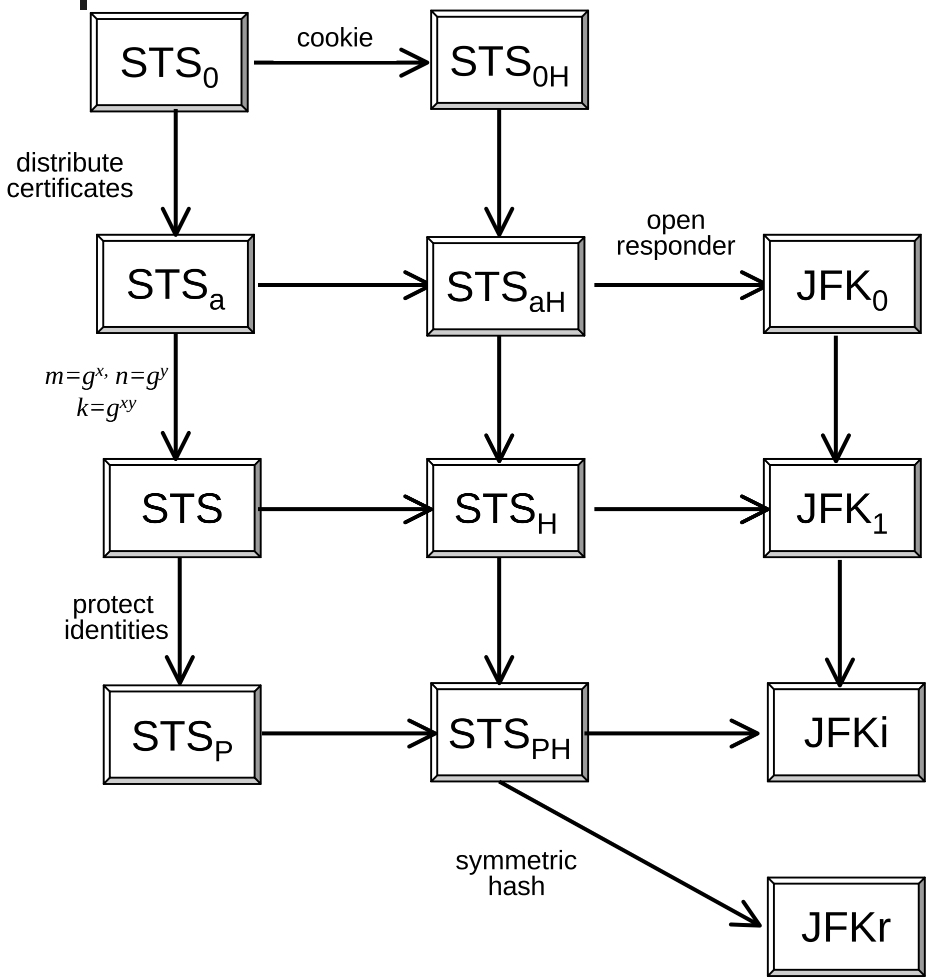
  - Shared secret: g^{ab}
  - Authenticated
  - Identity Protection
  - DoS Protection

Derivation Framework

- Protocols are constructed from:
  - components

  by applying a series of:
  - composition, refinement and transformation operations.
- Properties accumulate as a derivation proceeds.
- Examples in paper:
  - STS, ISO-9798-3, JFKi, JFKr, IKE

STS Family Derivation



Properties:

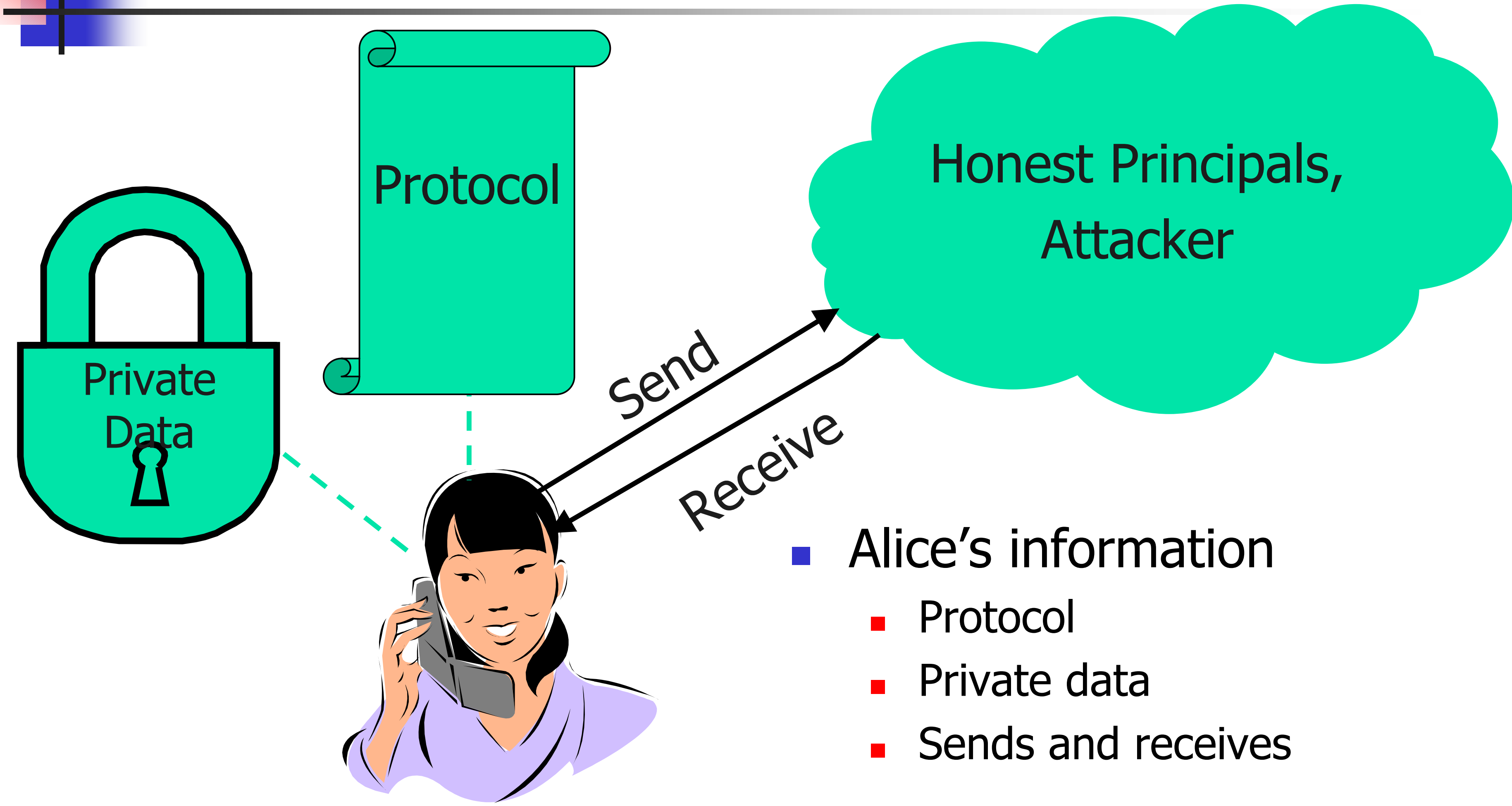
- Certificates from CA
- Shared secret: g^{ab}
- Identity protection
- DoS protection
- Reverse ID protection

Benefits and Directions

- Complex protocols are easier to understand and analyze.
- Protocols can be organized in a taxonomy.
  - e.g., STS family, Needham-Schroeder family.
- Protocol synthesis.

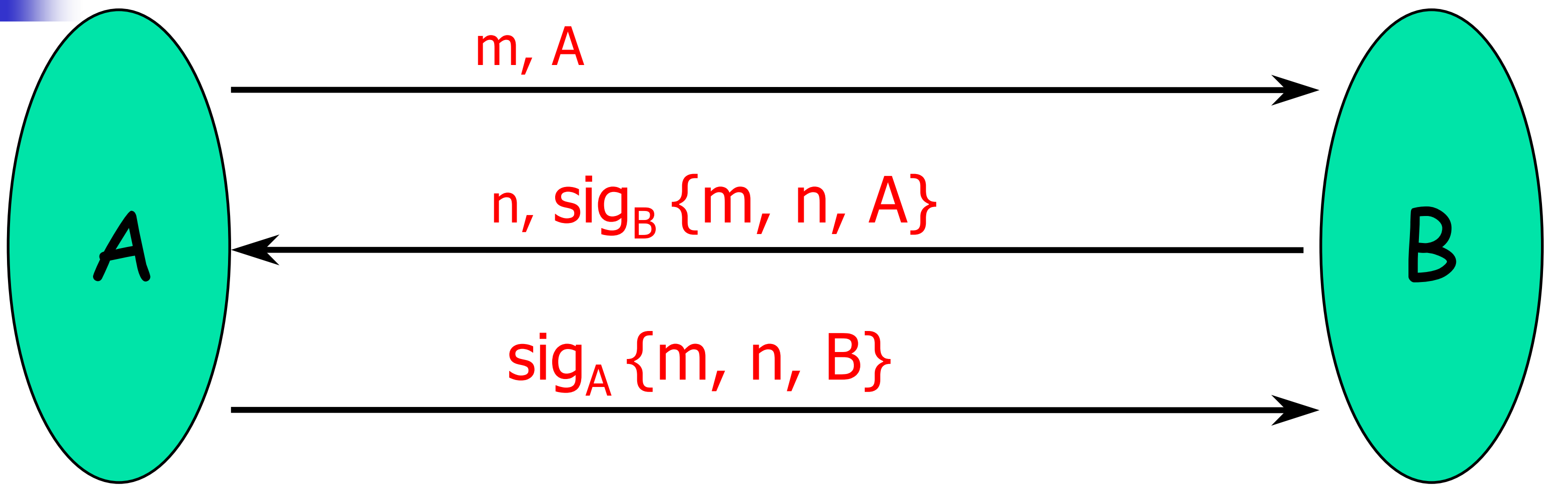
Compositional Logic

Protocol Logic: Main idea



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

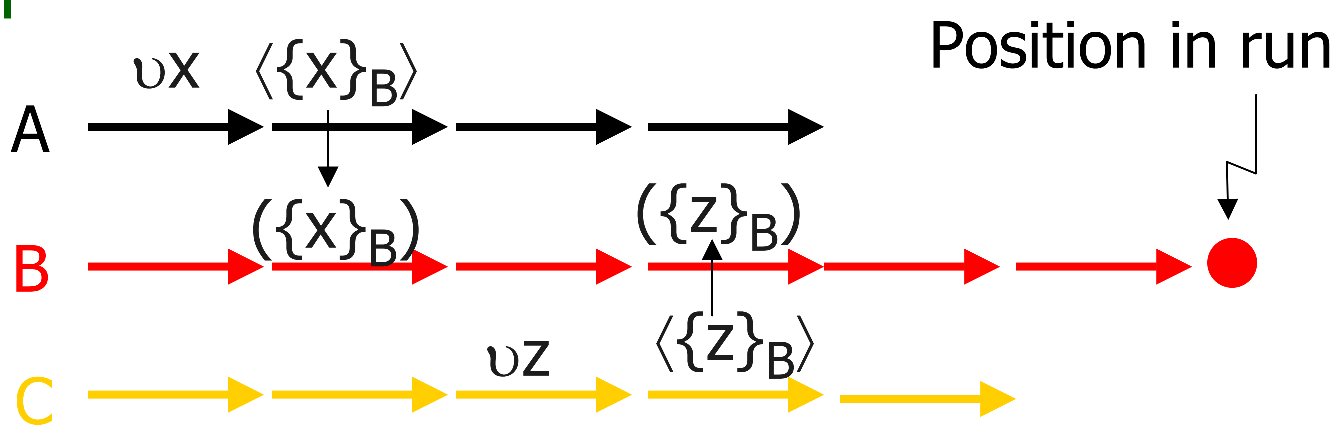
Example: Challenge-Response



- Alice reasons: if Bob is honest, then:
  - only Bob can generate his signature. [protocol independent]
  - if Bob generates a signature of the form sig_B {m, n, A},
    - he sends it as part of msg 2 of the protocol and
    - he must have received msg1 from Alice. [protocol specific]
- Alice deduces: Received (B, msg1) $\wedge$ Sent (B, msg2)

Execution Model

- Protocol
  - "Program" for each protocol role
- Initial configuration
  - Set of principals and key
  - Assignment of ≥1 role to each principal
- Run



Formulas true at a position in run

- Action formulas

  $a ::= \text{Send}(P,m) \mid \text{Receive}(P,m) \mid \text{New}(P,t)$
  $\mid \text{Decrypt}(P,t) \mid \text{Verify}(P,t)$

- Formulas

  $\varphi ::= a \mid \text{Has}(P,t) \mid \text{Fresh}(P,t) \mid \text{Honest}(N)$
  $\mid \text{Contains}(t_1, t_2) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x\ \varphi$
  $\mid \text{o}\varphi \mid \Diamond\varphi$

- Example

  $\text{After}(a,b) = \Diamond(b \wedge \text{o}\Diamond a)$

Modal Formulas

- After actions, postcondition

  $[\text{ actions }]_P\ \varphi$ where $P = \langle \text{princ, role id} \rangle$

- Before/after assertions

  $\varphi\ [\text{ actions }]_P\ \psi$

- Composition rule

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

Note: same P *in all formulas*

Diffie-Hellman: Property

- Formula
  - $[\ \text{new a}\ ]_A$ Fresh(A, g^a)

- Explanation
  - Modal form: $[\ \text{actions}\ ]_P\ \varphi$
  - Actions: $[\ \text{new a}\ ]_A$
  - Postcondition: Fresh(A, g^a)

Challenge Response: Property

- Modal form: φ [actions] P ψ
  - precondition: Fresh(A,m)
  - actions: [Initiator role actions] A
  - postcondition:

    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}}})

)

Composition: DH+CR = ISO-9798-3

- DH postcondition matches CR precondition
- Combination:
  - Substitute g^a for m in CR to obtain ISO.
  - Apply composition rule, persistence.
  - ISO initiator role inherits CR authentication.
- DH secrecy is also preserved
  - Proved using another application of composition rule.

Critical issues

- Reasoning about honest principals
  - Invariance rule, called “honesty rule”
- Preservation of invariants under composition
  - If we prove Honest(X) $\supset \varphi$ for protocol 1 and compose with protocol 2, is formula still true?

Honesty Rule

- Definition
  - A basic sequence of actions begins with receive, ends before next receive
- Rule

$$\frac{[\ ]_X\ \varphi \qquad \text{For all } B \in \text{BasicSeq}(Q).\ \varphi\ [B]_X\ \varphi}{Q \blacktriangleright \text{Honest}(X) \supset \varphi}$$

- Example

$$CR \blacktriangleright \text{Honest}(X) \supset (\text{Sent}(X, m_2) \supset \text{Recd}(X, m_1))$$

Combining protocols

$\overbrace{\text{DH} \blacktriangleright \text{Honest(X)} \supset \ldots}^{\Gamma}$ $\quad$ $\overbrace{\text{CR} \blacktriangleright \text{Honest(X)} \supset \ldots}^{\Gamma'}$

Γ |- Secrecy $\quad$ Γ' |- Authentication

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

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

DH $\bullet$ CR $\blacktriangleright$ $\Gamma \cup \Gamma'$

||

ISO $\blacktriangleright$ Secrecy $\wedge$ Authentication

Composition Rules

- Prove assertions from invariants

$$\Gamma \vdash \varphi\ [\ldots]_P\ \psi$$

- Invariant weakening rule

$$\frac{\Gamma \vdash \varphi\ [\ldots]_P\ \psi}{\Gamma \cup \Gamma' \vdash \varphi\ [\ldots]_P\ \psi}$$

If combining protocols, extend assertions to combined invariants

- Prove invariants from protocol

$$\frac{Q \blacktriangleright \Gamma \qquad Q' \blacktriangleright \Gamma}{Q \bullet Q' \blacktriangleright \Gamma}$$

Use honesty (invariant) rule to show that both protocols preserve assumed invariants

Conclusions and Future Work

Conclusions

- Protocol Derivation System:
  - Systematizes the practice of building protocols from standard sub-protocols. Useful for:
    - protocol analysis and understanding.
    - organizing related protocols in taxonomies.
    - protocol synthesis.
- Protocol Logic:
  - Correctness proofs follow derivation steps.
  - Rigorous treatment of protocol composition.

Future Work

- Derivation system:
  - taxonomies: STS, Needham-Schroeder family.
  - explore possibility of protocol synthesis.
  - can proofs in other formal systems be guided by derivations?
- Protocol Logic:
  - Formalize refinements and transformations.
  - Automate proofs.

Questions?