A Derivation System for Security Protocols and its Logical Formalization

Anupam Datta
John C. Mitchell

Ante Derek
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
  \[ \begin{align*}
  A & \rightarrow B: \ g^a \\
  B & \rightarrow A: \ g^b
  \end{align*} \]

- Shared secret (with someone)
  - A deduces:
    \[ \text{Knows}(Y, g^{ab}) \Rightarrow (Y = A) \lor \text{Knows}(Y, b) \]

- Authenticated
- Identity Protection
- DoS Protection
Component 2

- **Challenge Response:**
  
  $A \rightarrow B: \text{m, A}$
  $B \rightarrow A: \text{n, sig}_B \{\text{m, n, A}\}$
  $A \rightarrow B: \text{sig}_A \{\text{m, n, B}\}$

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

- ISO 9798-3 protocol:
  \[
  \begin{align*}
  A &\rightarrow B: \ g^a, A \\
  B &\rightarrow A: \ g^b, \text{sig}_B \{g^a, g^b, A\} \\
  A &\rightarrow B: \ \text{sig}_A \{g^a, g^b, B\}
  \end{align*}
  \]

- Shared secret: \(g^{ab}\)
- Authenticated
- Identity Protection
- DoS Protection
Refinement

- Encrypt signatures:
  
  $A \rightarrow B: g^a, A$
  
  $B \rightarrow A: g^b, E_K \{\text{sig}_B \{g^a, g^b, A\}\}$
  
  $A \rightarrow B: E_K \{\text{sig}_A \{g^a, g^b, B\}\}$

- 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$_{KB}$ {$g^b$, $g^a$}
  - $E_K$ {$\text{sig}_A$ {$g^a$, $g^b$, $B$}}
  - $B \rightarrow A$: $g^b$, $E_K$ {$\text{sig}_B$ {$g^a$, $g^b$, $A$}}

- 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

Private Data

Protocol

Honest Principals, Attacker

Send

Receive
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 \( \text{sig}_B \{m, n, A\} \),
  - he sends it as part of msg 2 of the protocol and
  - he must have received msg1 from Alice. [protocol specific]

Alice deduces: Received \((B, \text{msg1})\) \& Sent \((B, \text{msg2})\)
Execution Model

- **Protocol**
  - “Program” for each protocol role

- **Initial configuration**
  - Set of principals and key
  - Assignment of $\geq 1$ role to each principal

- **Run**

![Diagram showing the execution model with nodes A, B, and C, and arrows indicating interactions such as $\nu X \langle \{X\}_B \rangle$, $\nu Z \langle \{Z\}_B \rangle$]
Formulas true at a position in run

- **Action formulas**
  \[ a ::= \text{Send}(P,m) | \text{Receive}(P,m) | \text{New}(P,t) \]
  \[ \quad | \text{Decrypt}(P,t) | \text{Verify}(P,t) \]

- **Formulas**
  \[ \phi ::= a | \text{Has}(P,t) | \text{Fresh}(P,t) | \text{Honest}(N) \]
  \[ \quad | \text{Contains}(t_1, t_2) | \neg \phi | \phi_1 \land \phi_2 | \exists x \phi \]
  \[ \quad | \circ \phi | \Diamond \phi \]

- **Example**
  \[ \text{After}(a,b) = \Diamond(b \land \circ \Diamond a) \]
Modal Formulas

- After actions, postcondition
  \[ [\text{actions}]_P \varphi \]
  where \( P = \langle \text{princ}, \text{role id} \rangle \)

- Before/after assertions
  \[ \varphi \ [\text{actions}]_P \psi \]

- Composition rule
  \[
  \varphi [S]_P \psi \quad \psi [T]_P \theta
  \]
  \[ \varphi [ST]_P \theta \]

*Note: same \( P \) in all formulas*
Diffie-Hellman: Property

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

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

- Modal form: $\varphi \ [\text{actions}] P \psi$
  - precondition: $\text{Fresh}(A,m)$
  - actions: [Initiator role actions] $A$
  - postcondition:
    
    $\text{Honest}(B) \models \text{ActionsInOrder(}$
    
    $\text{send}(A, \{A,B,m\}),$
    
    $\text{receive}(B, \{A,B,m\}),$
    
    $\text{send}(B, \{B,A,\{n, \text{sig}_B \{m, n, A\}\}\}),$
    
    $\text{receive}(A, \{B,A,\{n, \text{sig}_B \{m, n, A\}\}\})$
    
)
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 $\text{Honest}(X) \supset \varphi$ 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**
  \[
  \text{For all } B \in \text{BasicSeq}(Q). \quad \varphi \ [B]_X \varphi \\
  Q \triangleright \text{Honest}(X) \supset \varphi
  \]

- **Example**
  \[
  \text{CR } \triangleright \text{Honest}(X) \supset \\
  (\text{Sent}(X, m_2) \supset \text{Recd}(X, m_1))
  \]
Combining protocols

\[ \Gamma \]

\[ \text{DH} \uparrow \text{Honest(X)} \supset \ldots \]

\[ \Gamma \vdash \text{Secrecy} \]

\[ \Gamma \cup \Gamma' \vdash \text{Secrecy} \]

\[ \Gamma \cup \Gamma' \vdash \text{Secrecy} \land \text{Authentication} \]

\[ \text{DH} \bullet \text{CR} \uparrow \Gamma \cup \Gamma' \]

\[ \parallel \]

\[ \text{ISO} \uparrow \text{Secrecy} \land \text{Authentication} \]

\[ \Gamma' \]

\[ \text{CR} \uparrow \text{Honest(X)} \supset \ldots \]

\[ \Gamma' \vdash \text{Authentication} \]

\[ \Gamma' \cup \Gamma' \vdash \text{Authentication} \]
Composition Rules

- Prove assertions from invariants
  \[ \Gamma \vdash \varphi […]_P \psi \]

- Invariant weakening rule
  \[ \frac{\Gamma \vdash \varphi […]_P \psi}{\Gamma \cup \Gamma' \vdash \varphi […]_P \psi} \]
  If combining protocols, extend assertions to combined invariants

- Prove invariants from protocol
  \[ Q \triangleright \Gamma \quad Q' \triangleright \Gamma \]
  \[ Q \cdot Q' \triangleright \Gamma \]
  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?