

Isabelle Implementation of Protocol Composition Logic

Dan Auerbach

Cary Kempston

Anupam Datta

Ante Derek

John C. Mitchell

Stanford University

October 22, 2004

Analysis of Protocols in PCL

1. Formalize the protocol: (Challenge-Response protocol)

$$[\text{Init}_{\text{CR}}]_X = [\text{new } x; \text{send } \hat{X}, \hat{Y}, x; \text{receive } \hat{Y}, \hat{X}, y, z; \\ \text{match } z / \text{SIG}_{\hat{Y}}(y, x, \hat{X})]_{\hat{X}}$$

2. State the Security Property: (Authentication)

$$\phi[\text{Init}_{\text{CR}}]_X \text{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y. \text{ActionsInOrder}(\\ \text{Send}(X, \hat{X}, \hat{Y}, x), \text{Receive}(Y, \hat{X}, \hat{Y}, x), \\ \text{Send}(Y, \hat{Y}, \hat{X}, y, \text{SIG}_{\hat{Y}}(y, x, \hat{X})), \\ \text{Receive}(X, \hat{Y}, \hat{X}, y, \text{SIG}_{\hat{Y}}(y, x, \hat{X})))$$

3. Axiomatically prove security properties.

Applied to practical protocols: GDOI [Meadows-Pavlovic04], IKEv2, SSL/TLS, 802.11i (WIP)

PCL Syntax

name ::= \hat{X}

thread ::= P

term ::= $x \mid name \mid ENC[K](t) \mid SIG[K](t)$

action ::= $\epsilon \mid \mathbf{send} \ t \mid \mathbf{receive} \ t \mid \mathbf{new} \ t \mid \mathbf{match} \ t/t$

strand ::= $strand ; action \mid action$

cord ::= $thread [strand]$

formula ::= $\mathbf{Send}(P, t) \mid \mathbf{Receive}(P, t) \mid \mathbf{New}(P, t) \mid \mathbf{Verify}(P, t) \mid$
 $\mathbf{Decrypt}(P, t) \mid \mathbf{Has}(P, t) \mid \mathbf{Fresh}(P, t) \mid \mathbf{Honest}(N) \mid$
 $\mathbf{FirstSend}(P, t, t') \mid \mathbf{Contains}(t_1, t_2) \mid \mathbf{Start}(P)$

modal ::= $\{formula, cord, formula\}$

PCL Axioms and Rules

AA1S $\phi[\text{send } t]_X \text{Send}(X, t)$

ORIG $\text{New}(X, x) \supset \text{Has}(X, x)$

P1N $\text{New}(X, t)[a]_X \text{New}(X, t)$

DEC $\text{Has}(X, \text{ENC}[K](x)) \wedge \text{Has}(X, K) \supset \text{Has}(X, x)$

SEC $\text{Honest}(\hat{X}) \wedge \text{Decrypt}(Y, \text{ENC}[\hat{X}](x)) \supset (\hat{Y} = \hat{X})$

$$\frac{\theta[P]_X \phi \quad \theta[P]_X \psi}{\theta[P]_X \phi \wedge \psi} \mathbf{G1} \qquad \frac{\theta[P]_X \phi \quad \theta' \supset \theta \quad \phi \supset \phi'}{\theta'[P]_X \phi'} \mathbf{G3}$$

$$\frac{\phi_1[P]_A \phi_2 \quad \phi_2[P']_A \phi_3}{\phi_1[PP']_A \phi_3} \mathbf{SEQ}$$

Isabelle

- Isabelle is a generic theorem-prover and logical framework [Paulson, 1989]
- Allows the implementation of new logics by specifying syntax and axioms.
- Better than implementing each logic from scratch; no need to supply methods for variable binding, rule instantiation, and proof.
- Structured proofs are available using interface packages Isar and ProofGeneral.

Encoding PCL in Isabelle

- The syntax and axioms are represented in a theory file:

```
consts
  PSend :: "[thread,CTerm] => o"
syntax
  PSend :: "[threadI,CTermlist] => actformI"
          ("Send' (_,_)")
axioms
  AA1S: "{P, X[send t], Send(X,t)}"
  REC  : "Receive(X,t) --> Has(X,t)"
  SEQ:  "[|{P, X[S1], Q} ; {Q, X[S2], R}|]
          ==> {P, X[S1 ; S2], R}"
```

Sample Proof (1)

```
lemma "{P,X[new t; send t],Has(X,t) &
        Send(X,t)}";
proof -;
have A: "{P,X[new t; send t],Has(X,t)}";
  apply (rule G3);
  apply (rule SEQ);
  apply (rule AA1N);
  apply (rule P1N);
  apply (blast);
  apply (rule ORIG);
done;
```

Sample Proof (2)

```
have B: "{P,X[new t; send t],Send(X,t)}";
  proof -;
  have C: "{P,X[new t],New(X,t)}" by (rule AA1N);
  have D: "{New(X,t),X[send t],Send(X,t)}"
    by (rule AA1S);
  from C D show "{P,X[new t; send t], Send(X,t)}"
    by (rule SEQ);
  qed;
from A B show "{P,X[new t; send t],
  Has(X,t) & Send(X,t)}"
  by (rule G1);
qed;
```

Current Status and Future Directions

- **Implementation:**
 - 322 line theory file: 200 lines of grammar, 122 lines of axioms and rules.
 - 45 axioms, 10 inference rules of PCL.
 - 10 auxiliary axioms (needed to work with Isabelle).
- **Future Directions:**
 - Automate(?) first order reasoning: interface with Isabelle's automatic deduction capabilities.
 - Investigate decidability of PCL.
 - Distribute PCL implementation to other users.
 - Machine-check proofs of properties of practical protocols like GDOI, IKEv2, SSL, 802.11i.