Computational Soundness for PCL

Dilsun Kaynar
Carnegie Mellon University

Foundations of Security and Privacy
October 11, 2007
Security protocol analysis

Protocol → Analysis Tool → Property → Attacker model

Security proof or attack
PCL Methodology

SSL

Protocol

Property

Attacker model

Security proof or attack

Axiomatic proof

Our tool: Protocol Composition Logic (PCL)

- Complete control over network
- Perfect crypto
Alternative view

- SSL
- Protocol
- Property
- Analysis Tool
- Attacker model
- Security proof or attack
- Associated proof techniques
- Alternative formalism
- Authentication
- Any ppt algorithm
- Explicit crypto
## Two worlds

<table>
<thead>
<tr>
<th></th>
<th>Symbolic model [NS78,DY84,...]</th>
<th>Complexity-theoretic model [GM84,...]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attacker actions</strong></td>
<td>- Fixed set of actions, e.g., decryption with known key (ABSTRACTION)</td>
<td>+ Any probabilistic poly-time computation</td>
</tr>
<tr>
<td><strong>Security properties</strong></td>
<td>- Idealized, e.g., secret message = not possessing atomic term representing message (ABSTRACTION)</td>
<td>+ Fine-grained, e.g., secret message = no partial information about bitstring representation</td>
</tr>
<tr>
<td><strong>Analysis methods</strong></td>
<td>+ Successful array of tools and techniques; automation</td>
<td>- Hand-proofs are difficult, error-prone; no automation</td>
</tr>
</tbody>
</table>

Can we get the best of both worlds?
PCL Approach

Protocol Composition Logic (PCL)
- Syntax
- Proof System

Symbolic “Dolev-Yao” model
- Semantics

Computational PCL
- Syntax $\pm \Delta$
- Proof System $\pm \Delta$

Complexity-theoretic model
- Semantics

Lectures on PCL so far

Leverage PCL success...
PCL Approach

High-level proof principles

PCL
- Syntax (Properties)
- Proof System (Proofs)

Symbolic Model
- PCL Semantics (Meaning of formulas)

Soundness Theorem (Induction)

Unbounded # concurrent sessions

Computational PCL
- Syntax ± \(\Delta\)
- Proof System ± \(\Delta\)

Soundness Theorem (Reduction)

Cryptographic Model
- PCL Semantics (Meaning of formulas)

 Polynomial # concurrent sessions

[BPW, MW,...]
Reminder: PCL Protocol Language

◆ Terms
  • Constants, variables, name, key, \((t,t)\), \(\text{sig}_K\{t\}\), \(\text{enc}_K\{t\}\)

◆ Actions
  • new \(t\), send \(t\), receive \(x\), sign\((t,K)\),
  • verify \((t,t,K)\), enc\((t,K)\), dec\((t,K)\)

◆ Strand, cord
  • \([a; ..., a]_p\)

◆ Role, protocol
InitCR(A, X) = [ new m; send A, X, {m, A}; receive X, A, {x, sig_B {m, n, 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}; ]
Attacker capabilities

- Controls complete network
  - Can read, remove, inject messages
- Fixed set of operations on terms
  - Pairing
  - Projection
  - Encryption with known key
  - Decryption with known key

Commonly referred to as “Dolev-Yao” attacker
**Execution model**

- **Initial configuration**
  - Protocol is a finite set of roles
  - Set of principals and keys
  - Assignment of $\geq 1$ role to each principal

- **Run (produces a symbolic trace)**

```plaintext
new x send \{x\}_B

A  \rightarrow B

receive \{x\}_B receive \{z\}_B

B  \rightarrow C

new z send \{z\}_B
```

C  \rightarrow A
Protocol Logic

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

- **Formulas**
  \[ \varphi ::= a | a < a | \text{Has}(P,t) | \text{Fresh}(P,t) | \text{Honest}(N) \]
  \[ | \text{Contains}(t, t) | \neg \varphi | \varphi \land \varphi | \exists x \varphi | \text{Start}(P) \]

- **Modal formulas**
  \[ \varphi [a; ...a]_P \varphi \]
Reasoning about knowledge

◆ Pairing
  • Has(X, {m,n}) ⊃ Has(X, m) ∧ Has(X, n)

◆ Encryption
  • Has(X, enc_k(m)) ∧ Has(X, K^{-1}) ⊃ Has(X, m)
Semantics

Protocol Q

- Defines set of roles (e.g., initiator, responder)
- Run R of Q is sequence of actions by principals following roles, plus attacker

Satisfaction  \( Q, R \models \phi \)

- \( Q, R \models \text{Send}(A, m) \)
  
  If in R, thread A executed send(m)

- \( Q, R \models \theta [ \text{actions} ]_P \phi \)
  
  If some role of P in R does exactly actions starting from state where \( \theta \) is true, then \( \phi \) is true in state after actions completed

- \( Q \models \theta [ \text{actions} ]_P \phi \)

  \( Q, R \models \theta [ \text{actions} ]_P \phi \) for all runs R of Q
Proof System

- **Goal:** formally prove security properties
- **Axioms**
  - Simple formulas provable by hand
- **Inference rules**
  - Proof steps
- **Theorem**
  - Formula obtained from axioms by application of inference rules
Soundness

Soundness Theorem:

• if $Q \vdash \phi$ then $Q \models \phi$

• If $\phi$ is a theorem then $\phi$ is a valid formula

• $\phi$ holds in any step in any run of protocol $Q$

• Unbounded number of participants

• Dolev-Yao intruder
Computational PCL

Symbolic proofs about complexity-theoretic model of cryptographic protocols!
Towards Computational PCL

◆ Revisit
  • Protocol language and execution model
  • Protocol logic

◆ New semantics for formulas:
  • Interpret formulas as operators on probability distributions on traces
  • Meaning of $\varphi$ on a set of traces $T$ is a subset $T'$ in which $\varphi$ holds. Probability that $\varphi$ holds is $|T'| / |T|$. 
Main result

✿ Computational PCL
  • Symbolic logic for proving security properties of network protocols using public-key encryption

✿ Soundness Theorem:
  • If a property is provable in CPCL, then property holds in computational model with overwhelming asymptotic probability.

✿ Benefits
  • Symbolic proofs about computational model
  • Computational reasoning in soundness proof (only!)
  • Different axioms rely on different crypto assumptions
Protocol execution model

- **Initialization**
  - Set roles, identify honest principals, provide encryption keys (polynomially-bounded) and random coins

- **Execution phase**
  - Adversary has complete control of the network, acts according to the standard cryptographic model
  - Running time is polynomially-bounded

- **Computational trace \(<e, ?>\)**
  - \(e\) is the symbolic description of the trace
  - \(?\) maps terms in \(e\) to bitstrings
PCL $\rightarrow$ Computational PCL

- Syntax, proof system mostly the same

- Significant difference
  - Symbolic "knowledge"
    - $\text{Has}(X,t)$: $X$ can produce $t$ from msgs that have been observed, by symbolic algorithm
  - Computational "knowledge"
    - $\text{Possess}(X,t)$: can produce $t$ by ppt algorithm
    - $\text{Indistinguishable}(X,t)$: can distinguish from random in ppt
  - More subtle system: some axioms rely on CCA2, some are info-theoretically true, etc.
Example: Simple secrecy

\[
\text{Init}(A, X) = [ \\
\text{new } m; \\
y := \text{enc}(m, X); \\
send (A, X, y); \\
]
\]

\[
\text{Resp}(B, X) = [ \\
\text{receive } z; \\
\text{match } z/ <X,B,z'>; \\
z'' := \text{dec}(z', B); \\
]
\]

When \( A \) completes proof with \( B \), \( m \) will be a shared secret:

\[
\text{Start}(A) [\text{Init}] \_A \text{ Honest}(A) \land \text{Honest}(B) \land (Z ? A,B) \Rightarrow \text{Indist}(Z,m)
\]
Example: axiomatic proof

? [new m] \( Z \neq X \Rightarrow \text{Indist}(Z, m) \)

- At the point \( m \) is generated by \( X \), it is indistinguishable from random by everyone else

\[ \text{Source}(Y, u, \{m\}_X) \land \neg \text{Decr}(X, \{m\}_X) \land \text{Honest}(X, Y) \land (Z \neq X) \Rightarrow \text{Indist}(Z, u) \]

- \( X \) remains indistinguishable since it appears encrypted with a private key
Central axioms

- **Cryptographic security property of encryption scheme**
  - Ciphertext indistinguishability

- **Cryptographic security property of signature scheme**
  - Unforgeability (used for authentication)
Sample axioms

- $\forall [\text{new } x] \forall Y \neq X \Rightarrow \text{Indist}(Y,x)$

- $\text{Source}(Y,u,\{m\}_x) \land \neg \text{Decr}(X, \{m\}_x) \land \text{Honest}(X,Y) \land (Z \neq X,Y) \Rightarrow \text{Indist}(Z,u)$

- $\text{Verify}(X,\text{sig}_Y\{m\}) \land \text{Honest}(X,Y) \Rightarrow \exists Y. \text{Sign}(Y,m)$
IND-CCA2-Secure encryption

IND-CCA2 security: ∀ PPT attackers A ∃ negligible function f ∃ n₀ ∀ security parameters n = n₀ Prob [d = b | A plays by the rules] <= ½ + f(n)

Intuition: Encryption reveals no information about message

Messages are bit-strings
Attacker is a PPT Turing Machine

Attacker wins if d = b
CMA-Secure Signatures

\[ \text{Challenger} \quad \text{mi} \quad \text{Sig}(Y,mi) \quad \text{Sig}(Y,m) \quad \text{Attacker} \]

Attacker wins if \( m \neq mi \)

\[ \text{SIG} \quad \text{Verify}(\hat{X}, SIG_Y \{ m \}) \land \text{Honest}(\hat{Y}) \supset \exists \bar{Y}. \text{Sign}(\bar{Y}, m) \]
Computational Semantics

- $[[\varphi]] (T,D,e)$
  - $T$ is a set of traces, $D$ is a Distinguisher,
  - $e$ is tolerance

- **Inductive definition**
  - $[[a(.,.)]] (T,D,e)$: for basic actions $a$, is simply the collection of traces in which action $a(.,.)$ occurs
  - $[[\text{Indist}(X,u)]] (T,D,e)$: uses $T$, $D$, $e$
    - $T$ if $D$ cannot distinguish $u$ from random, empty set otherwise
    - Not a trace property
Inductive Semantics

- $[[\varphi_1 \land \varphi_2]] (T,D,\varepsilon) = [[\varphi_1]] (T,D,\varepsilon) \cap [[\varphi_2]] (T,D,\varepsilon)$
- $[[\varphi_1 \lor \varphi_2]] (T,D,\varepsilon) = [[\varphi_1]] (T,D,\varepsilon) \cup [[\varphi_2]] (T,D,\varepsilon)$
- $[[\neg \varphi]] (T,D,\varepsilon) = T - [[\varphi]] (T,D,\varepsilon)$

**Implication uses conditional probability**

- $[[\varphi_1 \Rightarrow \varphi_2]] (T,D,\varepsilon) = [[\neg \varphi_1]] (T,D,\varepsilon)$
  $\cup [[\varphi_2]] (T',D,\varepsilon)$

where $T' = [[\varphi_1]] (T,D,\varepsilon)$

Formula defines transformation on probability distributions over traces
Complexity-theoretic semantics

\[ \text{\[Q\] \models \varphi \text{ if } \forall \text{ adversary } A \forall \text{ distinguisher } D \exists \text{ negligible function } f \exists n_0 \forall n > n_0 \text{ s.t.} \]

\[ [\varphi](T,D,f(n))/|T| > 1 - f(n) \]

- Fix protocol Q, PPT adversary A
- Choose value of security parameter n
- Vary random bits used by all programs
- Obtain set \(T = T(Q,A,n)\) of equi-probable traces
Soundness of Proof System

- Prove soundness of individual axioms
- Show all rules preserve validity
Soundness of proof system

◆ Example axiom
  • $\text{Source}(Y,u,\{m\}X) \land \neg \text{Decrypts}(X, \{m\}X) \land \text{Honest}(X,Y) \land (Z \neq X,Y) \Rightarrow \text{Indistinguishable}(Z, u)$

◆ Proof idea: crypto-style reduction
  • Assume axiom not valid:
    $\exists A \exists D \forall \text{negligible } f \forall n_0 \exists n > n_0 \text{ s.t.}$
    $[[\varphi]](T,D,f)/|T| < 1 - f(n)$
  • Construct attacker $A'$ that uses $A, D$ to break IND-CCA2 secure encryption scheme
Correspondence theorems

- If “secure” in abstract model, then “secure” in computational model.
  - Strong assumptions about cryptographic functions
  - No abstractions for Diffie-Hellman exponentiation
  - Negative results: symmetric encryption, hash functions, xor

- CPCL can potentially sidestep these problems by appropriate axiomatizations of properties
Logic and Cryptography: Big Picture

- Protocol security proofs using proof system
- Axiom in proof system
- Semantics and soundness theorem
- Complexity-theoretic crypto definitions (e.g., IND-CCA2 secure encryption)
- Crypto constructions satisfying definitions (e.g., Cramer-Shoup encryption scheme)