Protocol Verification by the Inductive Method

John Mitchell

Analysis Techniques

Recall: protocol state space

- Participant + attacker actions define a state transition graph
- A path in the graph is a trace of the protocol
- Graph can be
  - Finite if we limit number of agents, size of message, etc.
  - Infinite otherwise

Analysis using theorem proving

- Correctness instead of bugs
  - Use higher-order logic to reason about possible protocol executions
- No finite bounds
  - Any number of interleaved runs
  - Algebraic theory of messages
  - No restrictions on attacker
- Mechanized proofs
  - Automated tools can fill in parts of proofs
  - Proof checking can prevent errors in reasoning

Inductive proofs

- Define set of traces
  - Given protocol, a trace is one possible sequence of events, including attacks
- Prove correctness by induction
  - For every state in every trace, no security condition fails
  - Works for safety properties only
  - Proof by induction on the length of trace

Two forms of induction

- Usual form for \( \forall n \in \mathbb{N}. P(n) \)
  - Base case: \( P(0) \)
  - Induction step: \( P(x) \rightarrow P(x+1) \)
  - Conclusion: \( \forall n \in \mathbb{N}. P(n) \)
- Minimal counterexample form
  - Assume: \( \exists x \left[ \neg P(x) \land \forall y < x. P(y) \right] \)
  - Prove: contraction
  - Conclusion: \( \forall n \in \mathbb{N}. P(n) \)

Both equivalent to “the natural numbers are well-ordered”
Use second form

- Given set of traces
  - Choose shortest sequence to bad state
  - Assume all steps before that OK
  - Derive contradiction
    - Consider all possible steps

All states are good
Bad state

Sample Protocol Goals

- Authenticity: who sent it?
  - Fails if A receives message from B but thinks it is from C
- Integrity: has it been altered?
  - Fails if A receives message from B but message is not what B sent
- Secrecy: who can receive it?
  - Fails if attacker knows message that should be secret
- Anonymity
  - Fails if attacker or B knows action done by A

These are all safety properties

Inductive Method in a Nutshell

Informal Protocol Description
Abstract trace model
Correctness theorem about traces
Attacker inference rules

Try to prove the theorem

Theorem is correct

Work by Larry Paulson

- Isabelle theorem prover
  - General tool; protocol work since 1997
- Papers describing method
- Many case studies
  - Verification of SET protocol (6 papers)
  - Kerberos (3 papers)
  - TLS protocol
  - Yahalom protocol, smart cards, etc

http://www.cl.cam.ac.uk/users/lcp/papers/protocols.html

Isabelle

- Automated support for proof development
  - Higher-order logic
  - Serves as a logical framework
  - Supports ZF set theory & HOL
  - Generic treatment of inference rules
- Powerful simplifier & classical reasoner
- Strong support for inductive definitions
Agents and Messages

agent A,B,... = Server | Friend i | Spy
msg X,Y,... = Agent A
| Nonce N
| Key K
| \{ X, Y \}
| Crypt X K

Typed, free term algebra, ...

Protocol semantics

- Traces of events:
  - A sends X to B
- Operational model of agents
- Algebraic theory of messages (derived)
- A general attacker
- Proofs mechanized using Isabelle/HOL

Define sets inductively

- Traces
  - Set of sequences of events
  - Inductive definition involves implications
    if ev1, ..., evn ∈ evs, then add ev to evs
- Information from a set of messages
  - parts H : parts of messages in H
  - analz H : information derivable from H
  - synth H : msgs constructible from H

Protocol events in trace

- Several forms of events
  - A sends B message X
  - A receives X
  - A stores X

A→B \{A,N_A\}_{pk(B)}
If ev is a trace and N_A is unused, add
Says A B Crypt(pk B){A,N_A}
B→A \{N_B,N_A\}_{pk(A)}
If Says A' B Crypt(pk B){A,X} ∈ ev
and N_B is unused, add
Says B A Crypt(pk A){N_B,X}
A→B \{N_B\}_{pk(B)}
If Says ... (X,N_A) ... ∈ ev, add
Says A B Crypt(pk B){X}

Dolev-Yao Attacker Model

- Attacker is a nondeterministic process
- Attacker can
  - Intercept any message, decompose into parts
  - Decrypt if it knows the correct key
  - Create new message from data it has observed
- Attacker cannot
  - Gain partial knowledge
  - Perform statistical tests
  - Stage timing attacks, ...

Attacker Capabilities: Analysis

analz H is what attacker can learn from H

X ∈ H ⇒ X ∈ analz H
\{X,Y\} ∈ analz H ⇒ X ∈ analz H
\{X,Y\} ∈ analz H ⇒ Y ∈ analz H
Crypt X K ∈ analz H
& \ K \^2 ∈ analz H ⇒ X ∈ analz H
Attacker Capabilities: Synthesis

\[ \text{synth } H \text{ is what attacker can create from } H \]
\[ X \in H \implies X \in \text{synth } H \]
\[ X \in \text{synth } H & Y \in \text{synth } H \implies \{X, Y\} \in \text{synth } H \]
\[ X \in \text{synth } H & K \in \text{synth } H \implies \text{Crypt } XK \in \text{synth } H \]

Equations and implications

\[ \text{analz(analz } H) = \text{analz } H \]
\[ \text{synth(synth } H) = \text{synth } H \]
\[ \text{analz(synth } H) = \text{analz } H \cup \text{synth } H \]
\[ \text{synth(analz } H) = \text{???} \]

Nonce \( N \in \text{synth } H \implies \text{Nonce } N \in H \]
\[ \text{Crypt } KX \in \text{synth } H \implies \text{Crypt } KX \in H \]

Attacker and correctness conditions

If \( X \in \text{synth(analz(spies evs))}, \) add \textit{Says Spy } B \ X

\( X \) is not secret because attacker can construct it from the parts it learned from events.

If \textit{Says B A (N_b, X)_{pk(A)} } \in \text{evs} & \textit{Says A' B (N_a, Y)_{pk(B)} } \in \text{evs},
Then \textit{Says A B (N_a, Y)_{pk(B)} } \in \text{evs}

If B thinks he’s talking to A,
then A must think she’s talking to B.

Secure Electronic Transactions (SET)

◆ Cardholders and Merchants register
◆ They receive electronic credentials
  - Proof of identity
  - Evidence of trustworthiness
◆ Payment goes via the parties’ banks
  - Merchants don’t need card details
  - Bank does not see what you buy

Isabelle verification by
Larry Paulson, Giampaolo Bella, and Fabio Massacci

SET Certificate Hierarchy

Dual Signatures (idea used in SET)

◆ Link two messages sent to different receivers
◆ Each receiver can only read one message
  - Alice checks (message1, digest2, dual sig)
  - Bob checks (message2, digest1, dual sig)

\[ \text{DUAL SIGNATURE} \]
\[ \text{PRIVATE KEY} \]
\[ \text{SIGN NEW DIGEST WITH SIGNER'S PRIVATE KEY} \]
\[ \text{CONCATENATE DIGESTS TOGETHER} \]
\[ \text{HASH WITH SHA TO CREATE NEW DIGEST} \]
\[ \text{PRIVATE KEY} \]
\[ \text{SIGN NEW DIGEST WITH SIGNER'S PRIVATE KEY} \]
Verifying the SET Protocols

- Several sub-protocols
- Complex cryptographic primitives
- Many types of principals
  - Cardholder, Merchant, Payment Gateway, CAs
- Dual signatures: partial sharing of secrets
- 1000 pages of specification and description
- The upper limit of realistic verification

SET terminology

- Issuer
  - cardholder's bank
- Acquirer
  - merchant's bank
- Payment gateway
  - pays the merchant
- Certificate authority (CA)
  - issues electronic credentials
- Trust hierarchy
  - top CAs certify others

Players

- Issuing Bank
  - issues card
  - extends credit
  - assumes risk of card
  - cardholder reporting
- Merchant
  - merchant's bank
  - extends credit
  - assumes risk of merchant
  - funds merchant
- Consumer
- Processor
- Card Associations

SET Documentation

- Business Description
  - General overview
    - 72 pages
- Programmer's Guide
  - Message formats & English description of actions
    - 619 pages
- Formal Protocol Definition
  - Message formats & the equivalent ASN.1 definitions
    - 254 pages
- Total: 945 pages

The 5 sub-protocols of SET

- Cardholder registration
- Merchant registration
- Purchase request
- Payment authorization
- Payment capture
Cardholder Registration

- Two parties
  - Cardholder C
  - Certificate authority CA
- C delivers credit card number
- C completes registration form
  - Inserts security details
  - Discloses his public signature key
- Outcomes
  - C’s bank can vet the registration
  - CA associates C’s signing key with card details

SET messages

Message 5 in Isabelle

Secrecy of Session Keys

- Three keys, created for digital envelopes
  - Dependency: one key protects another
  - Main theorem on this dependency relation
  - Generalizes an approach used for simpler protocols (Yahalom)
- Similarly, prove secrecy of Nonces

Purchase Phase

Use SET Dual Signature

- 3-way agreement with partial knowledge
  - Cardholder shares Order Information (OI) only with Merchant
  - Cardholder shares Payment Information (PI) only with Payment Gateway
- Cardholder signs hashes of OI, PI
- Non-repudiation
  - All parties sign messages
Messages

The Purchase Request message

Complications in SET proofs

- Massive redundancy
  - Caused by hashing and dual signature
  - E.g. 9 copies of "purchase amount" in one message
- Multi-page subgoals
- Insufficient redundancy (no explicitness), failure of one agreement property
- Many digital envelopes

Inductive Method: Pros & Cons

- Advantages
  - Reason about infinite runs, message spaces
  - Trace model close to protocol specification
  - Can "prove" protocol correct
- Disadvantages
  - Does not always give an answer
  - Failure does not always yield an attack
  - Still trace-based properties only
  - Labor intensive
    - Must be comfortable with higher-order logic

Caveat

- Quote from Paulson (J Computer Security, 2000)
  The Inductive Approach to Verifying Cryptographic Protocols
  - The attack on the recursive protocol [40] is a sobering reminder of the limitations of formal methods. Making the model more detailed makes reasoning harder and, eventually, infeasible. A compositional approach seems necessary
- Reference