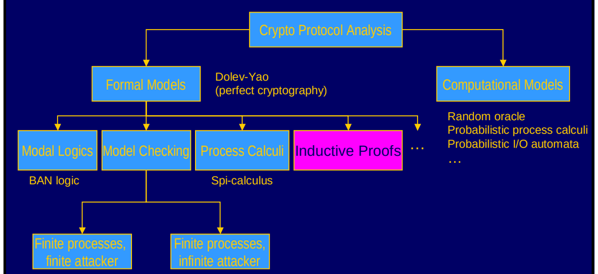
CS 259

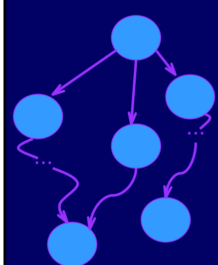
Protocol Verification by the Inductive Method

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

[Paulson]

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

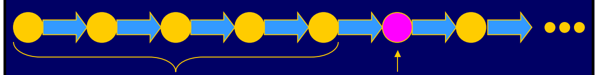
Two forms of induction

- Usual form for $\forall n \in Nat.\ P(n)$
  - Base case: $P(0)$
  - Induction step: $P(x) \Rightarrow P(x+1)$
  - Conclusion: $\forall n \in Nat.\ P(n)$
- Minimial counterexample form
  - Assume: $\exists x\ [\ \neg P(x) \wedge \forall y{<}x.\ P(y)\ ]$
  - Prove: contraction
  - Conclusion: $\forall n \in Nat.\ P(n)$

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

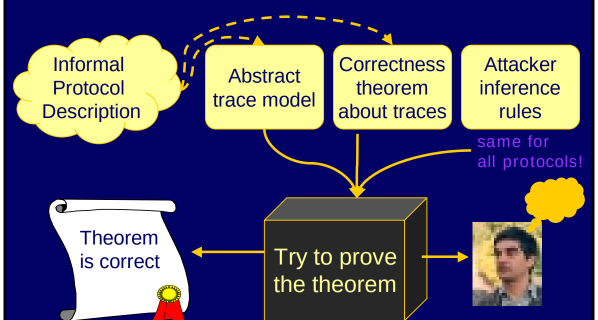


Sample Protocol Goals

- 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

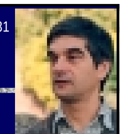
These are all safety properties

Inductive Method in a Nutshell



Work by Larry Paulson

Stanford Phd 1981



- 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 ev_1, …, $ev_n \in$ 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

$A \rightarrow B$ $\{A, N_A\}_{pk(B)}$ — If ev is a trace and Na is unused, add
`Says A B Crypt(pk B){A,Na}`

$B \rightarrow A$ $\{N_B, N_A\}_{pk(A)}$ — If **`Says A' B Crypt(pk B){A,X}`** $\in$ ev and Nb is unused, add
`Says B A Crypt(pk A){Nb,X}`

$A \rightarrow B$ $\{N_B\}_{pk(B)}$ — If **`Says ...{X,Na}...`** $\in$ 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*

$X \in H \Rightarrow X \in \text{analz } H$

$\{X, Y\} \in \text{analz } H \Rightarrow X \in \text{analz } H$

$\{X, Y\} \in \text{analz } H \Rightarrow Y \in \text{analz } H$

Crypt $X\,K \in \text{analz } H$
& $K^{-1} \in \text{analz } H \Rightarrow X \in \text{analz } H$

Attacker Capabilities: Synthesis

synth H is what attacker can create from H

infinite set!

$X \in H \Rightarrow X \in \text{synth } H$

$X \in \text{synth } H$ & $Y \in \text{synth } H$

$\Rightarrow \{X, Y\} \in \text{synth } H$

$X \in \text{synth } H$ & $K \in \text{synth } H$

$\Rightarrow \text{Crypt } X\, K \in \text{synth } H$

Equations and implications

analz(analz H) = analz H
synth(synth H) = synth H
analz(synth H) = analz H $\cup$ synth H
synth(analz H) = ???

Nonce $N \in$ synth H $\Rightarrow$ Nonce $N \in H$
Crypt $K\, X \in$ synth H $\Rightarrow$ Crypt $K\, X \in H$
or $X \in$ synth H & $K \in H$

Attacker and correctness conditions

If $X \in$ synth(analz(spies *evs*)),
add *Says Spy B X*

X is not secret because attacker can construct it from the parts it learned from *events*

If *Says B A* $\{N_b, X\}_{pk(A)} \in evs$ &
Says A' B $\{N_b\}_{pk(B)} \in evs$,
Then *Says A B* $\{N_b\}_{pk(B)} \in evs$

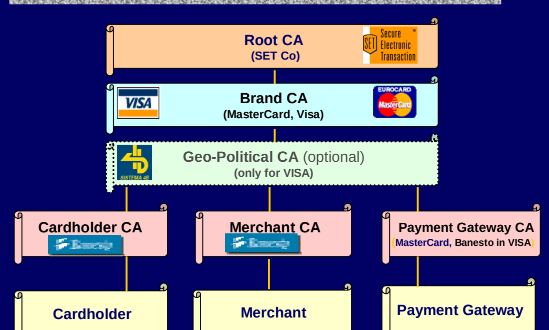
If B thinks he's talking to A, then A must think she's talking to B

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

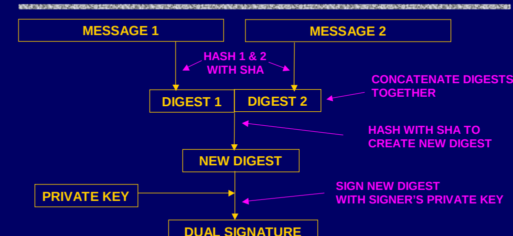
Isabelle verification by
Larry Paulson, Giampaolo Bella, and Fabio Massacci

SET Certificate Hierarchy



SOURCE: INZA.COM

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)

Verifying the SET Protocols

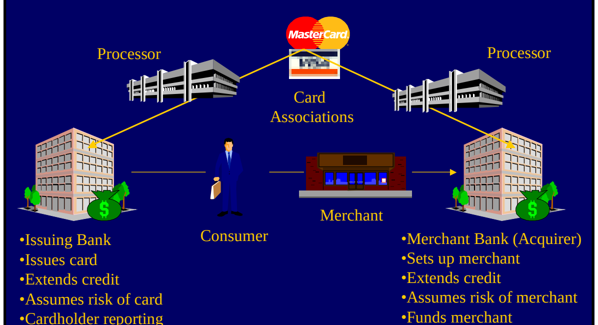
- Several sub-protocols
- Complex cryptographic primitives
- Many types of principals
  - Cardholder, Merchant, Payment Gateway, CAs
- Dual signatures: partial sharing of secrets
- 1000 pages of specification and description
- The upper limit of realistic verification

SET terminology

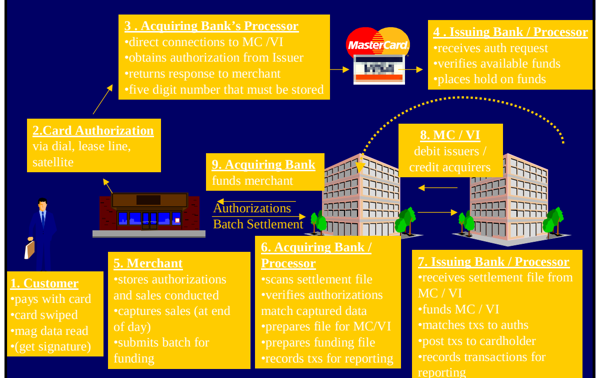
- Issuer
  - cardholder's bank
- Acquirer
  - merchant's bank
- Payment gateway
  - pays the merchant
- Certificate authority (CA)
  - issues electronic credentials
- Trust hierarchy
  - top CAs certify others

SOURCE: Michael I Shamos

Players



SOURCE: Michael I Shamos



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

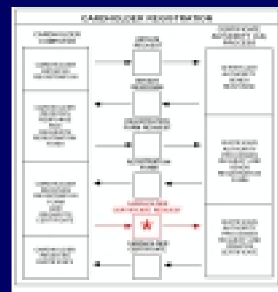
- Cardholder registration
- Merchant registration
- Purchase request
- Payment authorization
- Payment capture

Will look at these two briefly

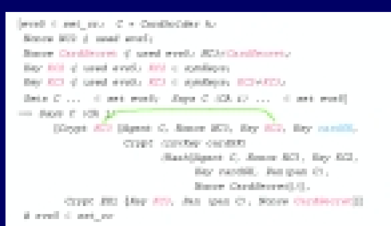
Cardholder Registration

- 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



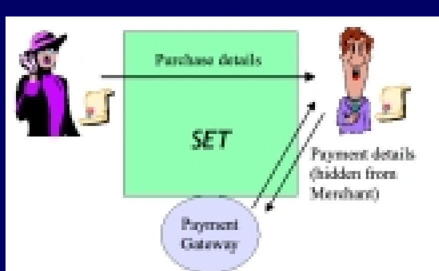
Message 5 in Isabelle



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

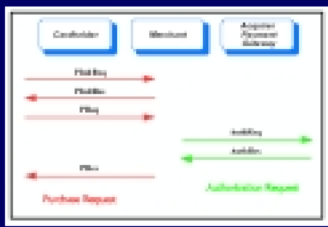
Purchase Phase



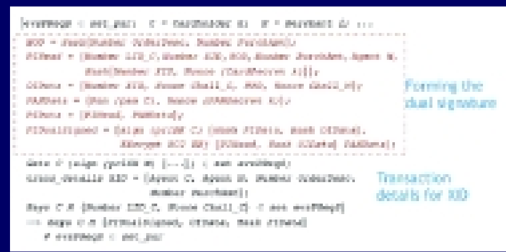
Use SET Dual Signature

- 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
  - [40] P.Y.A. Ryan and S.A. Schneider, An attack on a recursive authentication protocol: A cautionary tale. *Information Processing Letters* 65, 1 (January 1998) pp 7 – 10.