Verifying the SET Protocol

G Bella & L C Paulson  Cambridge
F Massacci  Siena
P Tramontano  Rome
Inductive Protocol Verification

- Define system’s operational semantics
- Include honest parties and an attacker
- Model each protocol step in an inductive definition
- Prove security properties by induction
- Mechanize using Isabelle/HOL
Can Big Protocols Be Verified?

• Can verify some real protocols:
  - Kerberos IV
  - TLS (the new version of SSL)
  - APM’s recursive protocol

• Other verification methods available:
  - Model-checking (Lowe)
  - NRL Protocol Analyzer (Meadows)
Growth in Protocol Complexity

- Needham-Schroeder (1978): 6 pages
- TLS: 80 pages
- SET: 5 main sub-protocols,
  3 manuals, nearly 1000 pages

Why so big?
Internet Shopping with SSL

Credit card details

SSL

Curses! Can’t get that number!
Do We Trust the Merchant?

Credit card details??

SSL

Now I can buy that software!
Do We Trust the Customer?

Fake card details

Send MS Office, charge to my card...
Basic Ideas of SET

- Legitimate Cardholders and Merchants receive electronic credentials
- Merchants don’t see credit card numbers (usually!)
- Payment is made via the parties’ banks
- Both sides are protected from fraud
SET Participants

- **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
Internet Shopping with SET

purchase details

SET

Her bank

His bank
SET Cryptographic Primitives

- Hashing, to make message digests
- Digital signatures
- Public-key encryption
- Symmetric-key encryption: session keys

- Digital envelopes involving all of these!
- Deep nesting of crypto functions
The 5 Sub-Protocols of SET

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

✓ verified!
Let’s look at this message.
Message 5 in Isabelle

[evs5 ∈ set_cr; C = Cardholder k;
Nonce NC3 ∉ used evs5;
Nonce CardSecret ∉ used evs5; NC3 ≠ CardSecret;
Key KC2 ∉ used evs5; KC2 ∈ symKeys;
Key KC3 ∉ used evs5; KC3 ∈ symKeys; KC2 ≠ KC3;
Gets C ... ∈ set evs5; Says C (CA i) ... ∈ set evs5]

⇒ Says C (CA i)

\{ Crypt KC3 \{ Agent C, Nonce NC3, Key KC2, Key cardSK, 
        Crypt (invKey cardSK) 
        (Hash\{ Agent C, Nonce NC3, Key KC2, 
            Key cardSK, Pan(pan C), 
            Nonce CardSecret\}) \}, 
    Crypt EKi \{ Key KC3, Pan (pan C), Nonce CardSecret \}\} 
# evs5 ∈ set_cr
What Did That Mean?

- Cardholder had asked to register a PAN (primary account number)
- Cardholder has received the CA’s reply
- Cardholder sends a digital envelope:
  - A public signing key, cardSK
  - A message, signed using the private key
  - *Two* session keys (one for the CA’s reply)
  - A secret number, CardSecret
Secrecy of the Card Number

• Intuitively obvious: PAN is always hashed or encrypted

• Huge case-splits caused by nested encryptions

• Two lemmas:
  - Session keys never encrypt PANs
  - Session keys never encrypt private keys
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)*
Secrecy of Nonces

- Secret numbers exchanged to generate Cardholder’s password
- Protected using those session keys
- Similar to the proofs for keys
- Main theorem about the Key/Nonce dependency relationship
The Purchase Phase!
Novel Aspects of SET Purchase

3-way agreement: with partial knowledge!

- Cardholder shares Order Information only with Merchant
- Cardholder shares Payment Information only with Payment Gateway
- Cardholder signs hashes of OI, PI
- Non-repudiation: all parties sign messages
Complications in SET Purchase

- Massive redundancy: exponential blow-ups
- Insufficient redundancy (no explicitness), requiring toil to prove trivial facts
- Two message flows: signed and unsigned
- Many digital envelopes
- No clear goals: What should I prove??
Conclusions

- Proofs are big, but not too big!
- Can prove secrecy for several keys and nonces, with dependency chains
- Can handle digital envelopes
- Merchant registration verified similarly—Purchase & Payment phases too!