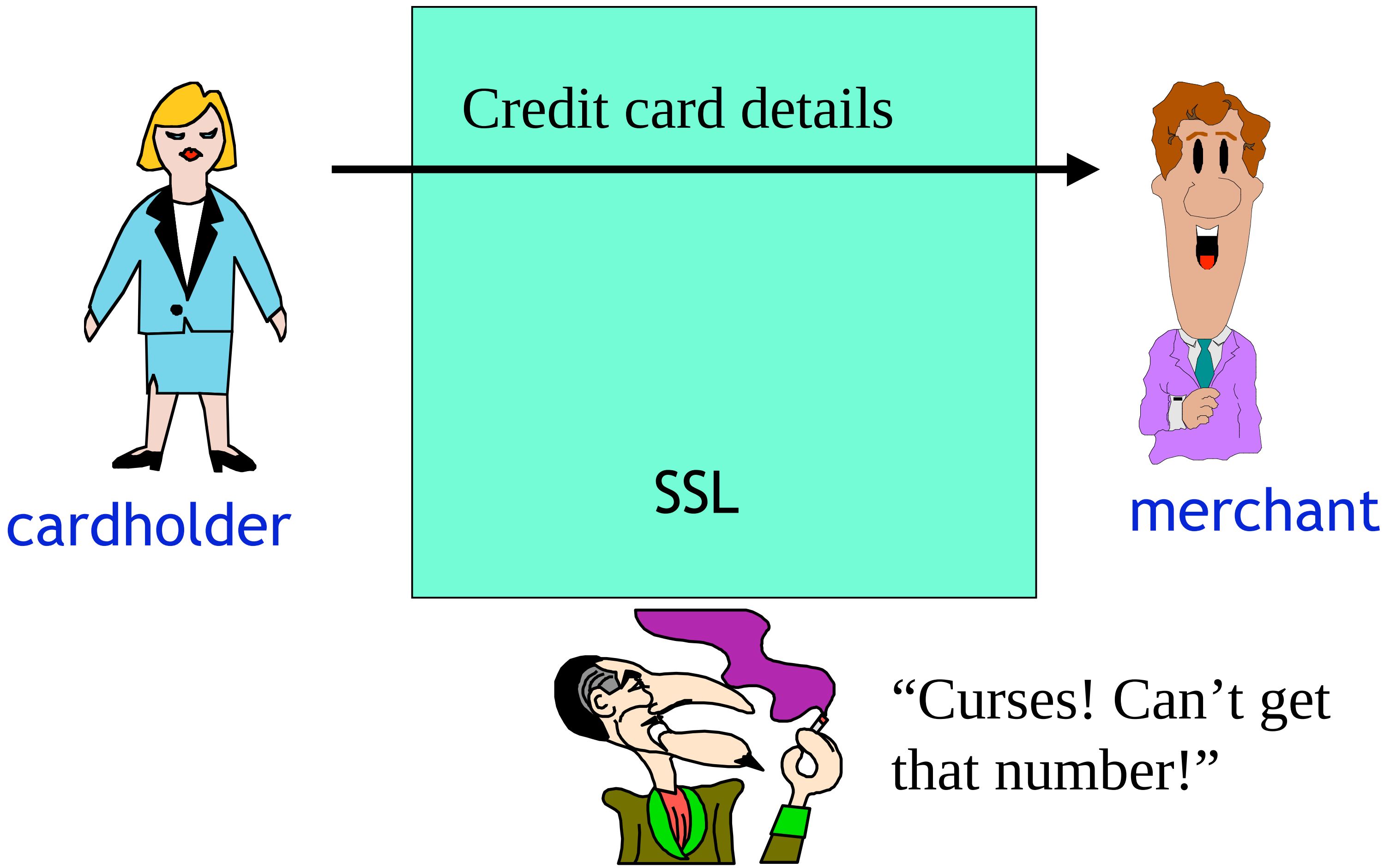
Verifying the SET Protocol: Overview

Lawrence C Paulson,
Computer Laboratory, University of Cambridge
(Joint with Giampaolo Bella and Fabio Massacci)

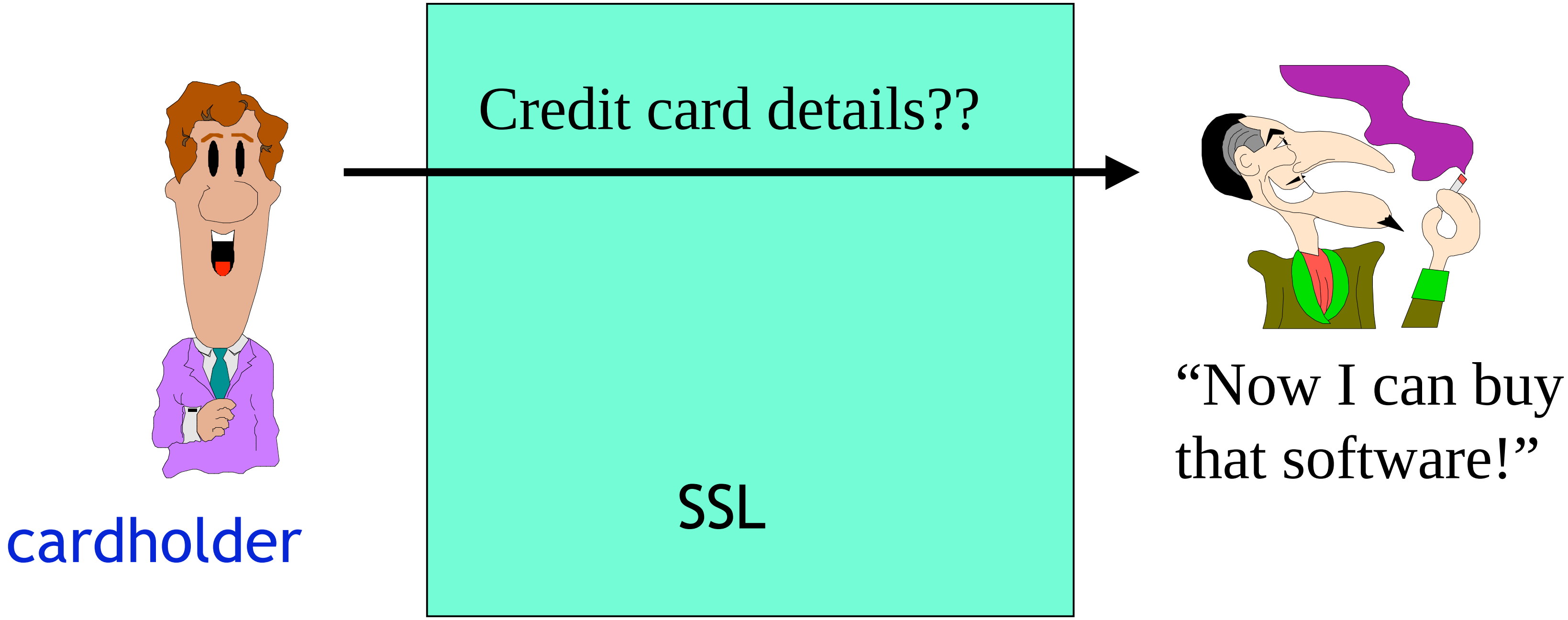
Plan of Talk

- The SET Protocol
- Defining the Formal Models
- Verifying the Registration Phase
- Verifying the Purchase Phase

Internet Shopping with SSL



Why Trust the Merchant?

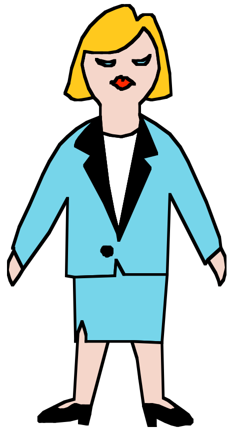


Why Trust the Customer?

"Send MS Office, charge to my card…"

Fake card details

SSL



merchant

Basic Ideas of SET

- Cardholders and Merchants must 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

Plan of Talk

- The SET Protocol
- Defining the Formal Models
- Verifying the Registration Phase
- Verifying the Purchase Phase

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

An Overview of Isabelle

- Generic: higher-order logic, set theory, …
- Good user interface (Proof General)
- Automatic document generation
- Powerful simplifier and classical prover
- Strong support for inductive definitions

The 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

SET Digital Envelopes

- Consisting of two parts:
  - Symmetric key K, encrypted with a public key
  - Main ciphertext, encrypted with K
- Hashing to link the two parts
- Minimal use of public-key encryption
- Great complications for formal reasoning
  - Numerous session keys in use
  - Dependency chains: keys encrypt keys

Obstacles to Formalization

- Huge size of documentation & protocol
- Lack of explicit objectives
- "Out of band" steps
- Many types of participants:
  - Cardholders
  - Merchants
  - Certificate Authorities
  - Payment Gateways (to pay merchants)

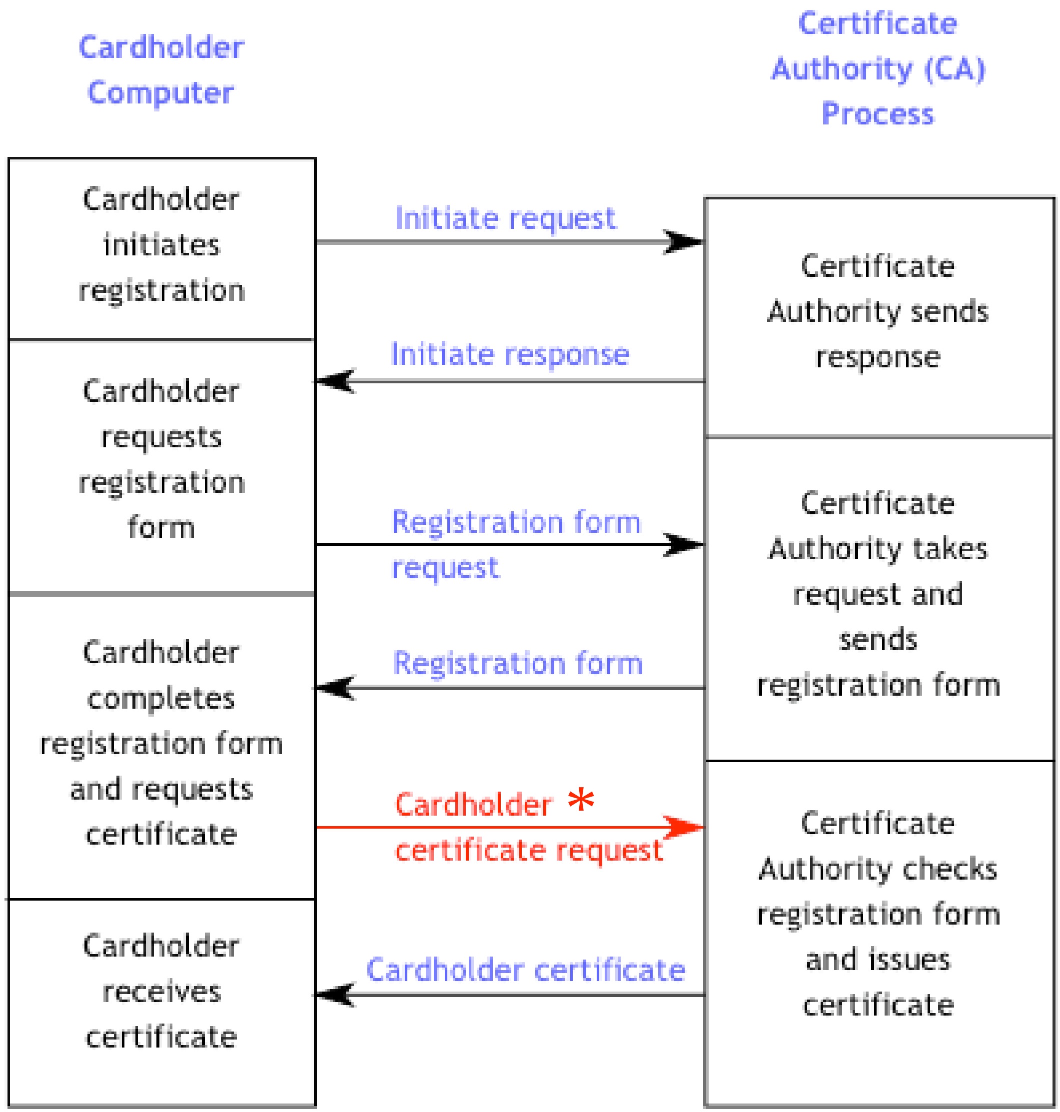
Plan of Talk

- The SET Protocol
- Defining the Formal Models
- Verifying the Registration Phase
- Verifying the Purchase Phase

Cardholder Registration

- Cardholder C and 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

Cardholder Registration



* 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
```

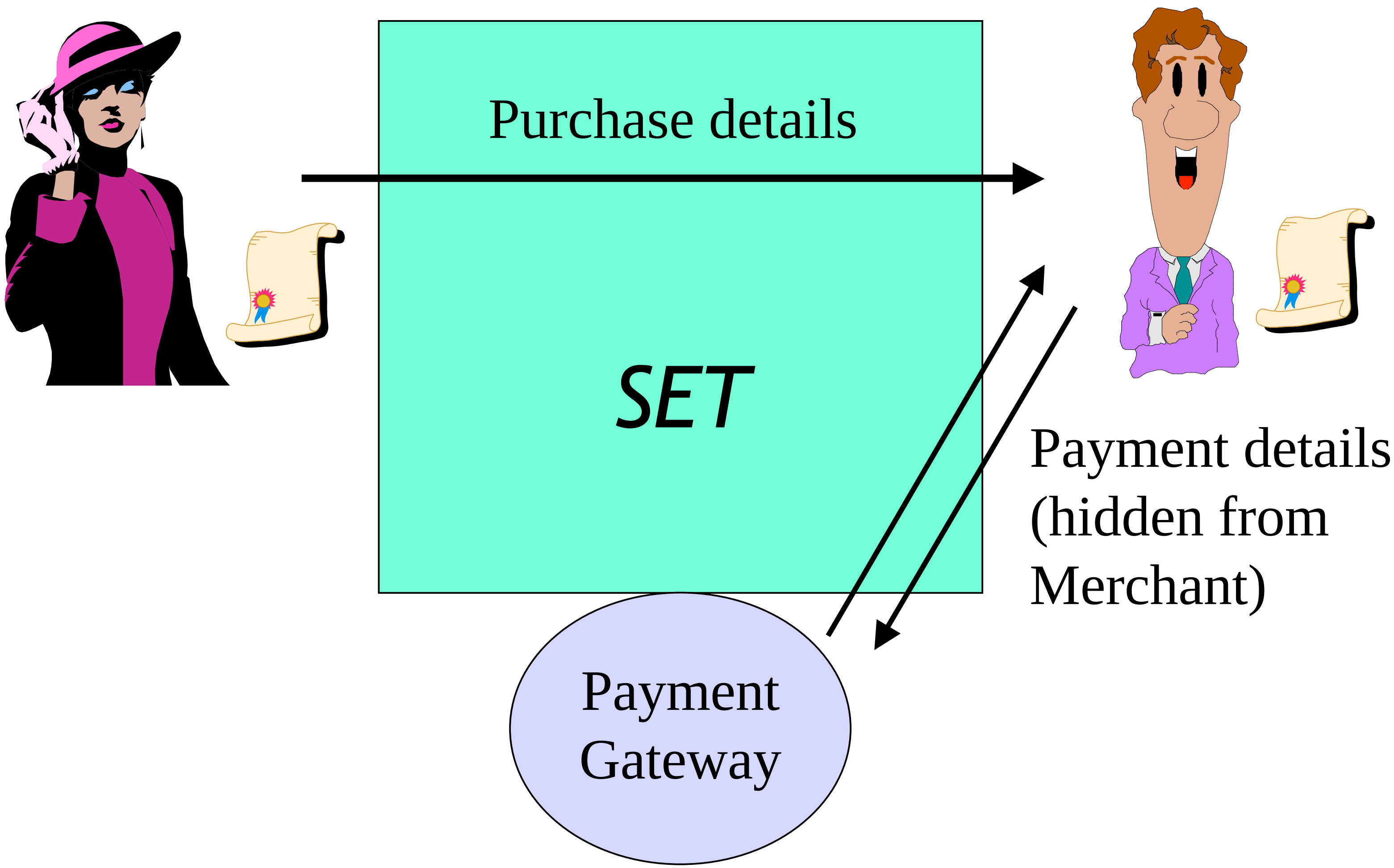
Secrecy of Session Keys

- 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

Plan of Talk

- The SET Protocol
- Defining the Formal Models
- Verifying the Registration Phase
- Verifying the Purchase Phase

The Purchase Phase



The SET Dual Signature

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

The *Purchase Request* Message

```
[|evsPReqS ∈ set_pur;  C = Cardholder k;  M = Merchant i; ...
  HOD = Hash{|Number OrderDesc, Number PurchAmt|};
  PIHead = {|Number LID_C, Number XID, HOD, Number PurchAmt, Agent M,
            Hash{|Number XID, Nonce (CardSecret k)|}|};
  OIData = {|Number XID, Nonce Chall_C, HOD, Nonce Chall_M|};
  PANData = {|Pan (pan C), Nonce (PANSecret k)|};
  PIData = {|PIHead, PANData|};
  PIDualSigned = {|sign (priSK C) {|Hash PIData, Hash OIData|},
                  EXcrypt KC2 EKj {|PIHead, Hash OIData|} PANData|};
  Gets C (sign (priSK M) {|...|}) ∈ set evsPReqS;
  trans_details XID = {|Agent C, Agent M, Number OrderDesc,
                        Number PurchAmt|};
  Says C M {|Number LID_C, Nonce Chall_C|} ∈ set evsPReqS|]
  ⟹ Says C M {|PIDualSigned, OIData, Hash PIData|}
     # evsPReqS ∈ set_pur
```

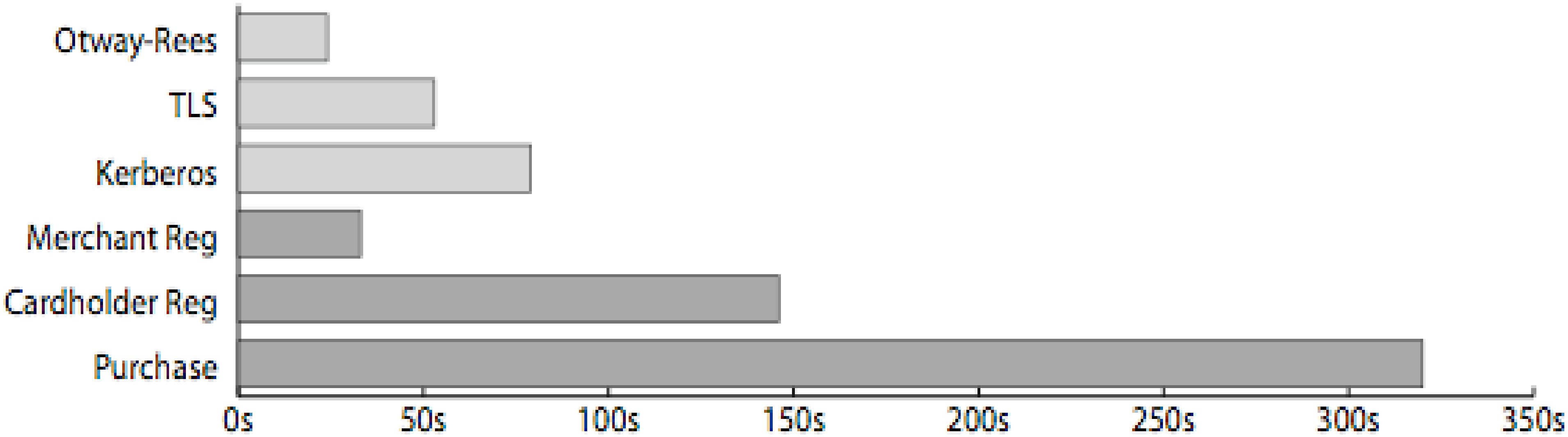
Forming the dual signature

Transaction details for XID

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

Runtimes for Various Protocols



Conclusions

- We can find flaws in massive protocols
- Analyzing bigger protocols than SET may be impossible
- Improvements are needed:
  - Abstract treatment of constructions such as digital envelopes
  - Better official formal definitions