# Verification of SET:
# The Purchase Phase

G Bella & L C Paulson *Cambridge*
F Massacci *Siena*

# Overview of the Model

- Traces of events
  - $A$ sends $B$ message $X$
  - $A$ receives $X$
  - $A$ stores $X$
- A powerful attacker
  - is an accepted user
  - attempts all possible splicing attacks
  - has the same specification in all protocols

# Agents and Messages

$$agent \quad A, B, \ldots = \text{Server} \mid \text{Friend}\, i \mid \text{Spy}$$

$$\begin{aligned} message \quad X, Y, \ldots &= \text{Agent}\, A \\ &\mid \text{Nonce}\, N \\ &\mid \text{Key}\, K \\ &\mid \{X, X'\} \qquad \text{compound message} \\ &\mid \text{Crypt}\, K\, X \end{aligned}$$

free algebras: we assume PERFECT ENCRYPTION

# Maps over Message Sets

- parts $H$: message components

  $$\text{Crypt}\, K\, X \;\mapsto\; X$$

- analz $H$: accessible components

  $$\text{Crypt}\, K\, X,\; K^{-1} \;\mapsto\; X$$

- synth $H$: expressible messages

  $$X,\; K \;\mapsto\; \text{Crypt}\, K\, X$$

RELATIONS are traditional, but FUNCTIONS give us an equational theory

# The Function $\mathsf{analz}\, H$

$$\frac{\mathsf{Crypt}\, K\, X \in \mathsf{analz}\, H \qquad K^{-1} \in \mathsf{analz}\, H}{X \in \mathsf{analz}\, H}$$

$$\frac{X \in H}{X \in \mathsf{analz}\, H} \qquad \frac{\{X, Y\} \in \mathsf{analz}\, H}{X \in \mathsf{analz}\, H} \qquad \frac{\{X, Y\} \in \mathsf{analz}\, H}{Y \in \mathsf{analz}\, H}$$

Typical derived law:

$$\mathsf{analz}\, G \cup \mathsf{analz}\, H \subseteq \mathsf{analz}(G \cup H)$$

# A Few Equations

$\mathsf{parts}(\mathsf{parts}\,H) = \mathsf{parts}\,H$ transitivity

$\mathsf{analz}(\mathsf{synth}\,H) = \mathsf{analz}\,H \cup \mathsf{synth}\,H$ "cut elimination"

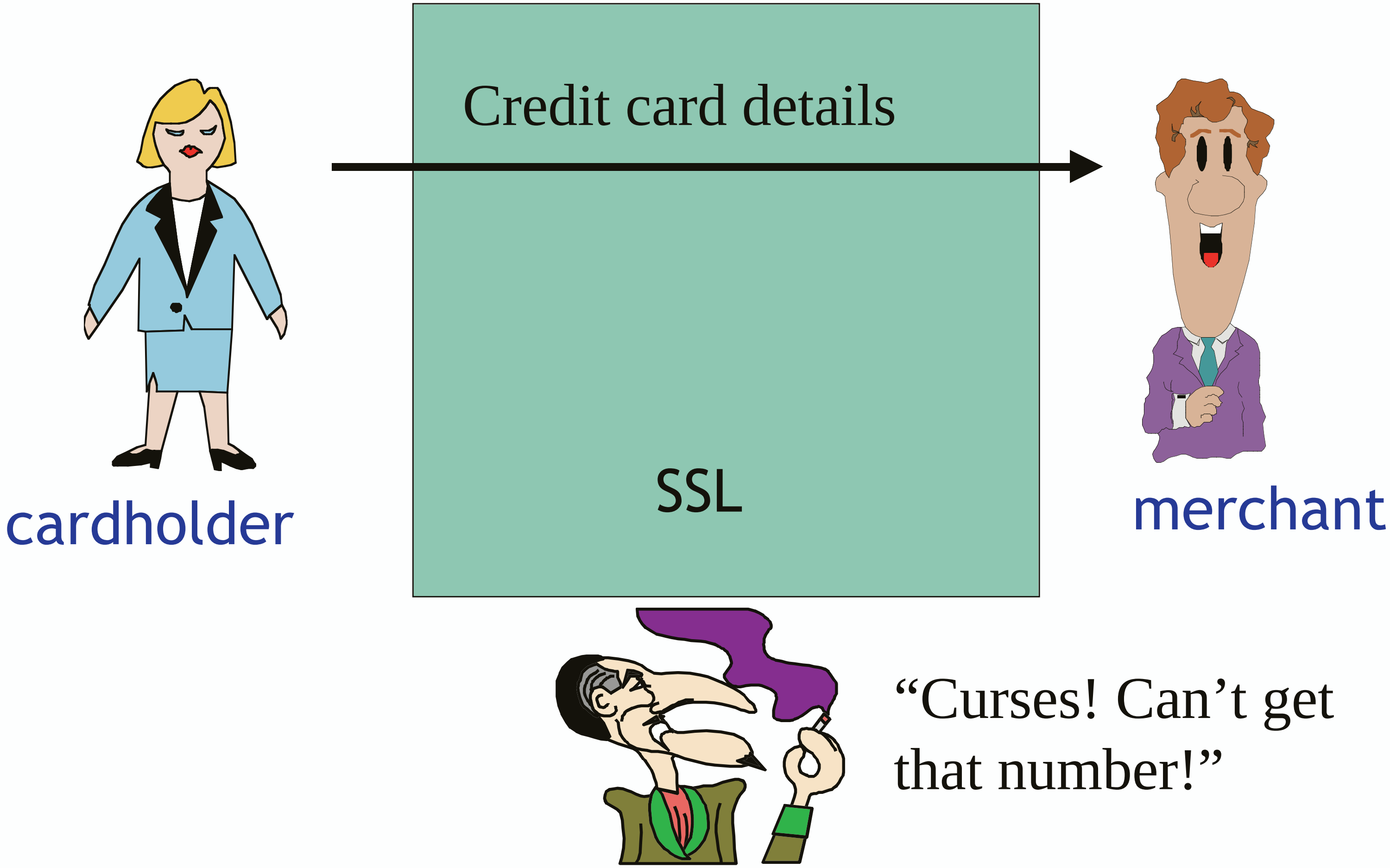
Symbolic Evaluation:

$$\mathsf{analz}(\{\mathsf{Crypt}\,K\,X\} \cup H) =$$
$$\begin{cases} \{\mathsf{Crypt}\,K\,X\} \cup \mathsf{analz}(\{X\} \cup H) & \text{if } K^{-1} \in \mathsf{analz}\,H \\ \{\mathsf{Crypt}\,K\,X\} \cup \mathsf{analz}\,H & \text{otherwise} \end{cases}$$

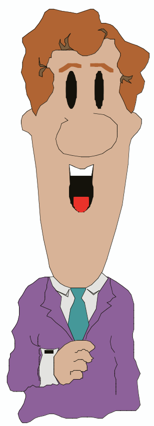
# Can Big Protocols Be Verified?

- Can verify some real protocols:
  - Kerberos IV
  - TLS (the latest version of SSL)
  - APM's recursive protocol
- Other verification methods available:
  - Model-checking (Lowe)
  - NRL Protocol Analyzer (Meadows)
  - Many others (you!)

# Internet Shopping with SSL



# Do We Trust the Merchant?



cardholder

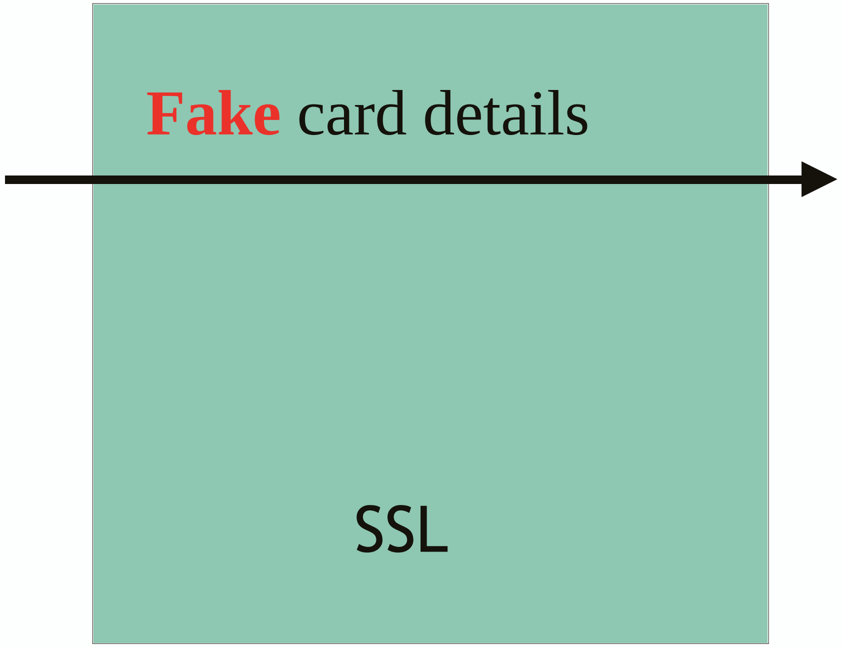
Credit card details??

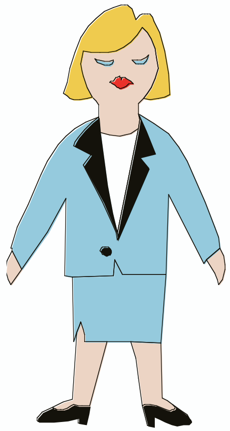
SSL

"Now I can buy that software!"

# Do We Trust the Customer?

“Send MS Office, charge to my card…”



merchant

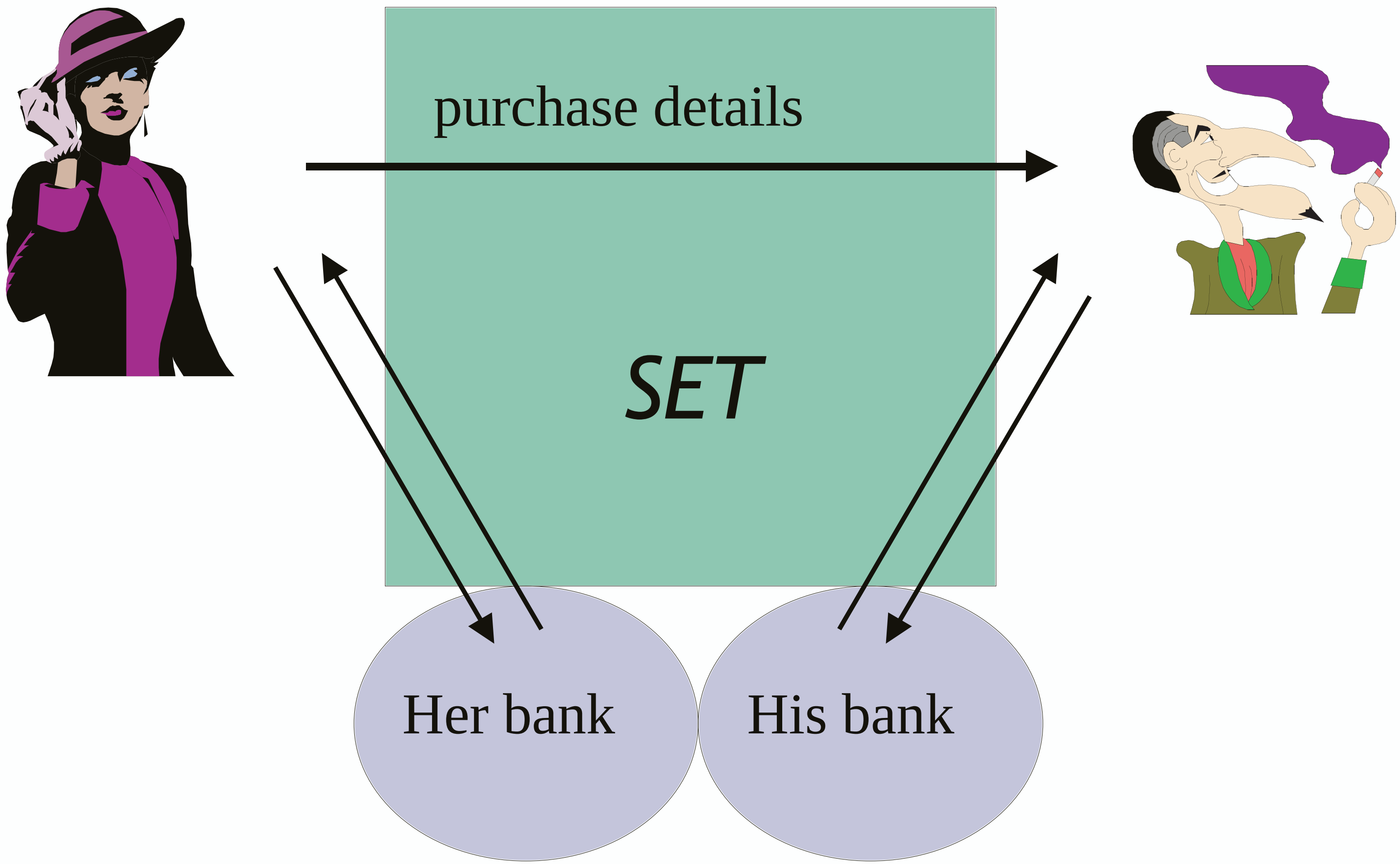
# Basic Ideas of SET

- Legitimate Cardholders and Merchants receive electronic credentials
- Merchants don't need credit card numbers
- 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 credentials
- Trust hierarchy: top CAs certify others

# Internet Shopping with SET



# SET Cryptographic Primitives

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

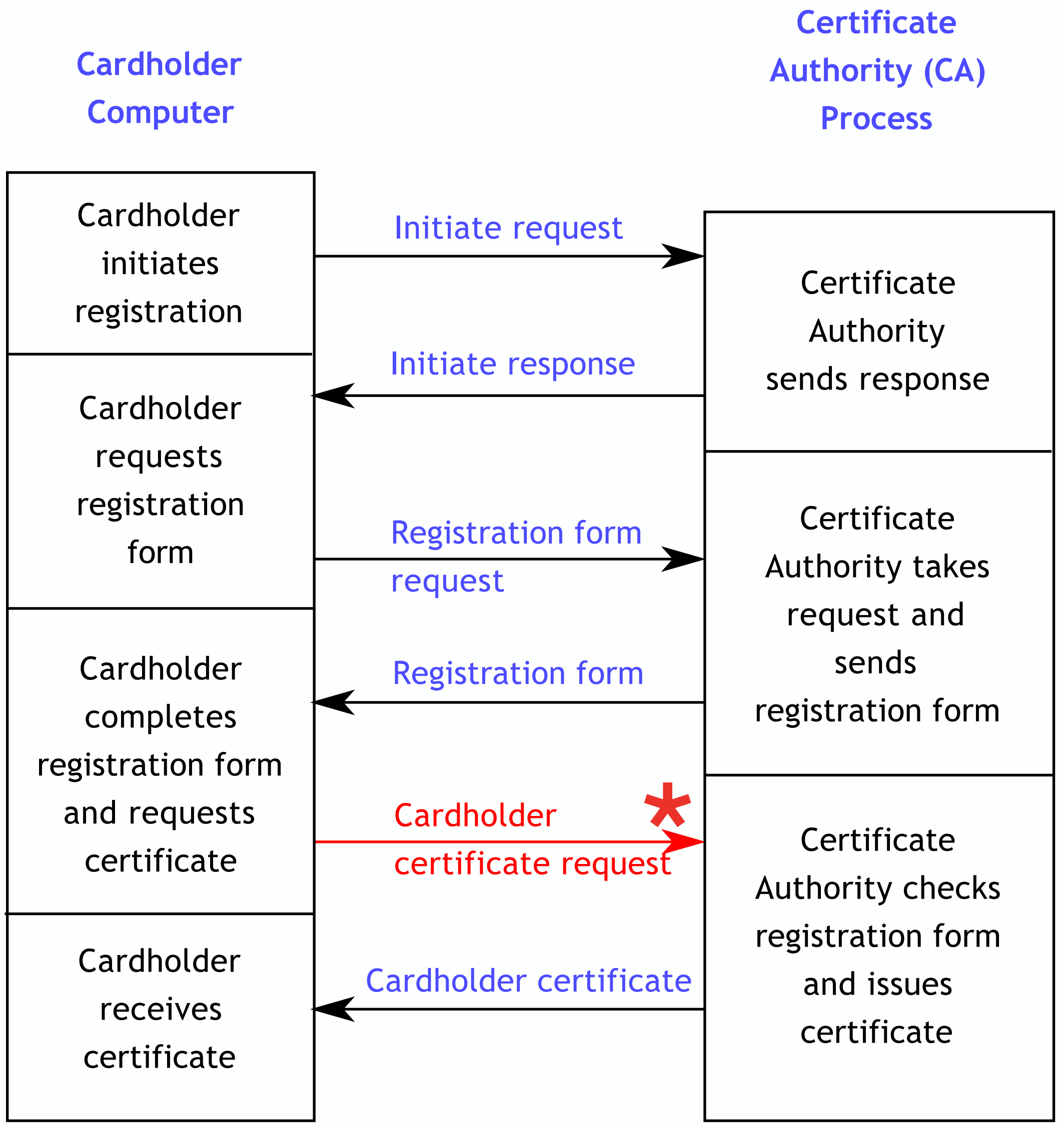
- Digital envelopes involving all of these!
- Deep nesting of crypto functions

# The 5 Sub-Protocols of SET

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

✓ *verified!* (whatever that means)

# Cardholder Registration




* Let’s look at this message

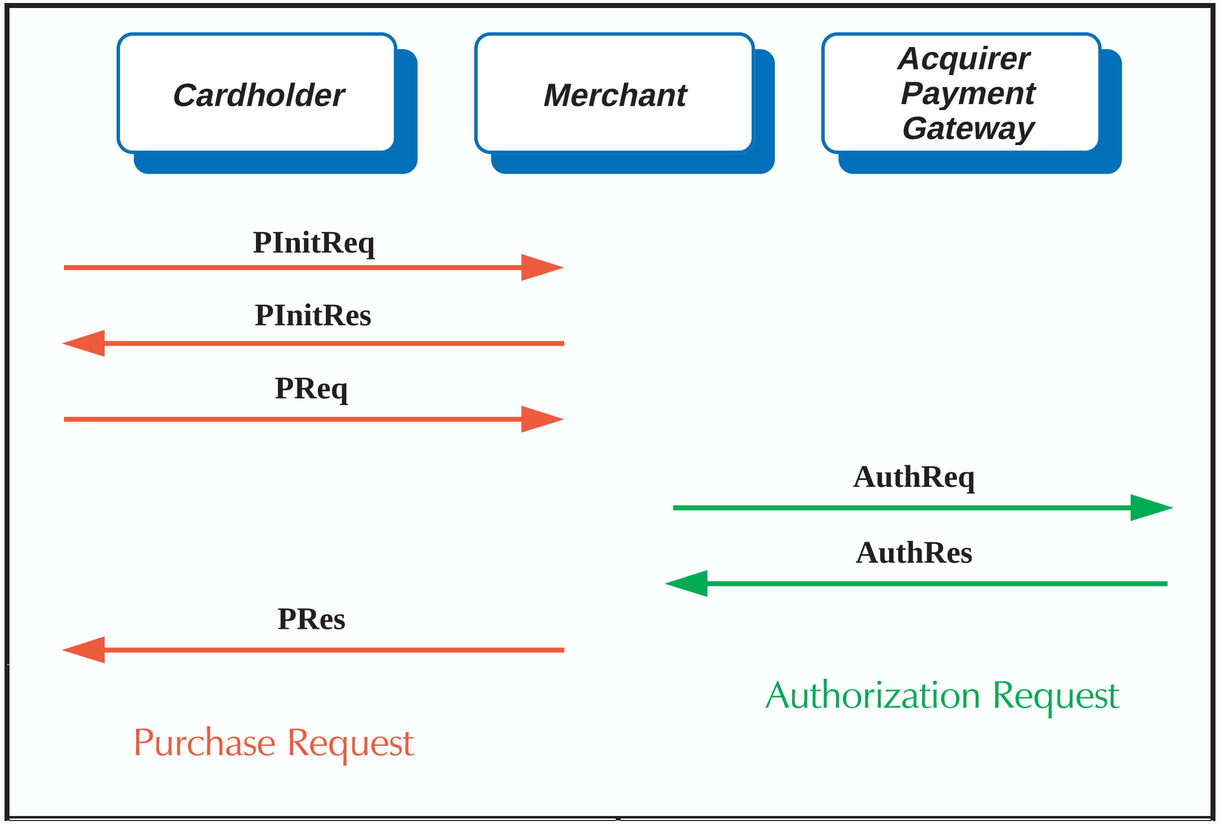
# Message 5 in Isabelle

```
⟦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

# The Purchase Phase!



# Purchase Request in Isabelle

```
⟦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

# 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?