Smart-Card Protocols and E-Commerce Protocols

Giampaolo Bella

Overview

- Smart card protocols
  - Shoup-Rubin.
  - The outcomes of *provable security*.
  - Goal availability vs. explicitness.
- E-commerce protocols
  - SET and its controversy.
  - cardholder registration in SET.

Where to keep long-term secrets?

With traditional protocols...

- Workstations are not reliable (e.g. Trojan horse attacks).
- Users are not reliable (e.g. accidents, conspiracies, dictionary attacks).

With smart card protocols...

- Are PINs secure?
- Are smart cards tamper-resistant?
- Do they really strengthen the protocol goals?

Formal analysis of smart protocols

1. **Authentication logics** [Abadi, Burrows, Kaufman, Lampson, 1990]
   - \+ short, abstract proofs
   - − dubious soundness
   - − costly expressiveness enhancement
2. **Provable security** [Bellare & Rogaway, 1995] [Shoup & Rubin, 1996]
   - \+ theoretical, sound approach
   - − design gist hard to grasp
   - − design must adapt to approach
3. **Inductive approach ?**

What's a smart card?

A microcomputer!

- Currently 8-bit processors.
- A single serial port for I/O.
- Cheaper and cheaper; more and more popular (GSM phones, pre-payed gas meters, recent O/S's, etc.).

Its ROM provides a *secure shell* for secrets!

Secure shell?

Non-invasive techniques.

- Fault generation.
  - External radiations (e.g. Biham and Shamir on DES).
  - Power or clock supply glitches (e.g. Anderson and Kuhn on Pay-TV systems).

Invasive techniques.

- Chip disembedding chemicals (from Chemistry!).
- Laser-cutter microscopes (from Electrical Engineering!).
- Microprobing needles (from Cellular Biology!).

Example

A fault generation attack from Pay-TV hacking . . .

```
1  b = answer_address
2  a = answer_length
3  if (a == 0) goto 8
4  transmit(*b)
5  b = b + 1
6  a = a - 1
7  goto 3
8  ...
```

Exercise: find a good glitch to break 6!

Shoup-Rubin's contribution, 1996

1. Analyse variant of Leighton-Micali's key distribution protocol (1993, no smart cards) by Bellare-Rogaway approach (1995).
2. Extend protocol and approach with smart cards.
3. Analyse new protocol (implemented lately) verifying that
   - peers share same session key (no formal treatment);
   - session key is *confidential* (proof by hand).

   Practicality?

Leighton-Micali's secret-key agreement

– $\boxed{\mathsf{Pairkey}(A, B) = \{\!|A|\!\}_{Kb} \oplus \{\!|B|\!\}_{Ka}}$ $\leadsto$ Π_{ab}

Pairkeys are calculated by the server and sent unencrypted.

– $\boxed{\mathsf{pairK}(A, B) = \{\!|A|\!\}_{Kb}}$ $\leadsto$ π_{ab}

B's card calculates the pairk for A and B directly;

A's card does so upon reception of the pairkey.

The spy *knows* some pairkeys and some pairk's.

The Shoup-Rubin protocol — phases I, II, II

I.	1.	A	$\to$	S	:	A, B
	2.	S	$\to$	A	:	$\Pi_{ab}, \{\!|\Pi_{ab}, B|\!\}_{Ka}$
II.	3.	A	$\to$	C_a	:	A
	4.	C_a	$\to$	A	:	$Na, \{\!|Na|\!\}_{K_{Ca}}$
III.	5.	A	$\to$	B	:	A, Na

The Shoup-Rubin protocol — phases IV-VII

IV.	6.	B	$\to$	C_b	:	A, Na
	7.	C_b	$\to$	B	:	$Nb, Kab, \{\|Na, Nb\|\}_{\pi_{ab}}, \{\|Nb\|\}_{\pi_{ab}}$
V.	8.	B	$\to$	A	:	$Nb, \{\|Na, Nb\|\}_{\pi_{ab}}$
VI.	9.	A	$\to$	C_a	:	$B, Na, Nb, \Pi_{ab}, \{\|\Pi_{ab}, B\|\}_{Ka}, \{\|Na, Nb\|\}_{\pi_{ab}}, \{\|Na\|\}_{K_{Ca}}$
	10.	C_a	$\to$	A	:	$Kab, \{\|Nb\|\}_{\pi_{ab}}$
VII.	11.	A	$\to$	B	:	$\{\|Nb\|\}_{\pi_{ab}}$

The significance of authenticators

- $\{|\Pi_{ab}, B|\}_{Ka}$ tells A the pairkey's peer.
  ($\Pi_{ab} = \{|A|\}_{Kb} \oplus \{|B|\}_{Ka}$)
- $\{|Na|\}_{K_{Ca}}$ saves A's card RAM.
- $\{|Na, Nb|\}_{\pi_{ab}}$ associates the two nonces.
- $\{|Nb|\}_{\pi_{ab}}$ serves for authenticating A with B.

All encrypted under long-term keys!

An obvious risk

$$Kab = \{\!|Nb \cdot 0 \cdot 0|\!\}_{\pi_{ab}}$$

$$(\Pi_{ab} = \{\!|A|\!\}_{Kb} \oplus \{\!|B|\!\}_{Ka}; \qquad \pi_{ab} = \{\!|A|\!\}_{Kb})$$

Both Π_{ab} and Nb *are* available to the spy because sent in clear.

Session key forged if either B's card's broken
or A's card's broken.

Obvious but realistic!

Goals achieved by Shoup-Rubin

- Strong authenticity and unicity.
- Confidentiality, key distribution, authentication?

IV.	6.	B	$\rightarrow$	C_b	:	A, Na
	7.	C_b	$\rightarrow$	B	:	$Nb, Kab, \{\!|Na, Nb|\!\}_{\pi_{ab}}, \{\!|Nb|\!\}_{\pi_{ab}}$
			$\vdots$			
VI.	9.	A	$\rightarrow$	C_a	:	$B, Na, Nb, \Pi_{ab}, \{\!|\Pi_{ab}, B|\!\}_{Ka}, \{\!|Na, Nb|\!\}_{\pi_{ab}}, \{\!|Na|\!\}_{KCa}$
	10.	C_a	$\rightarrow$	A	:	$Kab, \{\!|Nb|\!\}_{\pi_{ab}}$
VII.	11.	A	$\rightarrow$	B	:	$\{\!|Nb|\!\}_{\pi_{ab}}$

Peers implicit in messages 7 and 10!

The peers' viewpoints

B's . . . 7. $C_b \to B$: $Nb, Kab, \mathit{Cert1}, \mathit{Cert2}$

$\vdots$

11. $A \to B$: $\mathit{Cert2}$

A's . . . 10. $C_a \to A$: $Kab, \mathit{Cert2}$

No peer can associate Kab if cards' data buses are eavesdropped.

On goal availability

Confidentiality (and others) can be proved if certificates are inspected.

For example:

if A receives

$$Kab, \{\!|Nb|\!\}_{\pi_{ab}}$$

neither A nor B are the spy, and neither A's nor B's card are cloned, then Kab is confidential.

The goal is not available to A.

Adding explicitness to Shoup-Rubin

$$
\begin{array}{llllll}
\text{IV.} & 6. & B & \rightarrow & C_b & : A, Na \\
 & 7. & C_b & \rightarrow & B & : Nb, A, Kab, \{\!|Na, Nb|\!\}_{\pi_{ab}}, \{\!|Nb|\!\}_{\pi_{ab}} \\
 & & & \vdots & & \\
\text{VI.} & 9. & A & \rightarrow & C_a & : B, Na, Nb, \Pi_{ab}, \{\!|\Pi_{ab}, B|\!\}_{Ka}, \\
 & & & & & \quad \{\!|Na, Nb|\!\}_{\pi_{ab}}, \{\!|Na|\!\}_{K_{Ca}} \\
 & 10. & C_a & \rightarrow & A & : B, Kab, \{\!|Nb|\!\}_{\pi_{ab}} \\
 & & & & & \\
\text{VII.} & 11. & A & \rightarrow & B & : \{\!|Nb|\!\}_{\pi_{ab}}
\end{array}
$$

Confidentiality, key distribution, authentication now available to peers.

E-commerce protocols

The SET controversy

In 1997 the Mastercard/VISA experts *hurry up* to ship the

Secure Electronic Transactions (SET)

family of protocols because E-commerce can't wait.

- Ambiguities, contradictions, omissions.
  - "Options" are not optional!
  - Informal text often corrects definitions!
  - Certain messages can be handled at discretion!
  - Certain nonces are issued but not used!
- Vague specification of the goals.

SET documentation

1. Business description.
2. Formal protocol definition.
3. Programmer's guide.

Nearly 1000 pages! Cyberlaw?

Ambiguities? An example.

There is a difference between non-required and optional. Non-required fields may be omitted according to the SET protocol. Optional fields may be omitted according to ASN.1 encoding rules. In some messages, a field may be optional according to ASN.1, but still required by the SET protocol. In these cases, it is incumbent on the application to fill in these fields. [Loeb, 1998].

How to implement SET?

Giving freedom to applications is dangerous!

Goal vagueness? Some examples.

In cardholder registration phase.

- Can cardholder C register more than one key with the same certification authority?
- Can cardholders C and C' register the same key with the same certification authority?
- Are credit card numbers confidential info?
  From Programmer's Guide: YES.
  From Business Description: NO, payment gateways transmit cardholder data in clear.

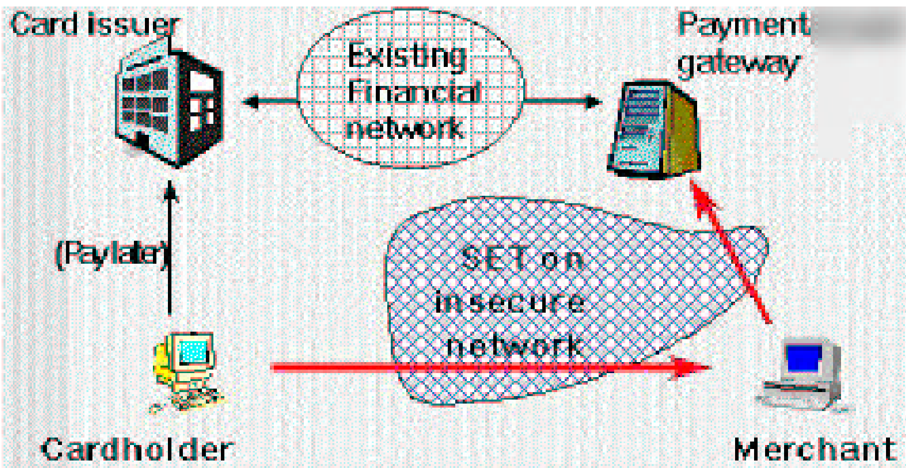
Will SET ever take off?

Some researchers say NO.

- 33 new products (by Hitachi, IBM, VeriSign, etc.) using SET are being tested.
- Microsoft Wallet uses SET.
  Easier E-commerce:
  1. each user submits credit-card number and billing and shipping address to a Microsoft server.
  2. Shopping at one of the 50+ adhering sites becomes a single click.

We need formal analysis either way!

Basic SET layout



The cardholder doesn't trust the merchant!

Technicalities

The cardholder C does trust the payment gateway PG.

So, C sends M his info packaged as

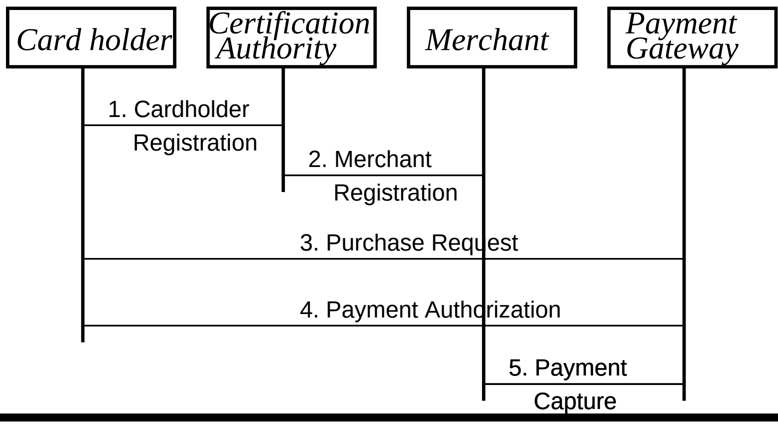
$$\{\!|C, M, \{\!|C\_info|\!\}_{K\mathrm{pg}}|\!\}_{Kc^{-1}}$$

The merchant M trusts PG.

A combination of symmetric and asymmetric crypto achieves the goal.

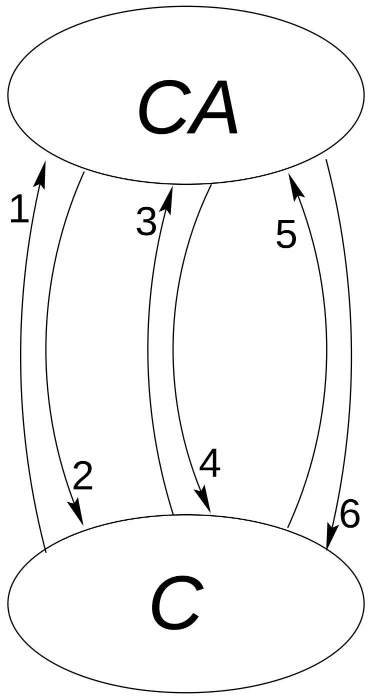
Basic Components of SET

- cardholders.
- Rooted hierarchy of Certification Authorities.
- Merchants.
- Payment Gateways.
- 5 phases.



cardholder registration in SET (abstract)

1. **Initiate Request**. C initiates sending identity and fresh nonce.
2. **Initiate Response**. CA quotes the received data and attaches a certificate with his *public key*.
3. **Registration Form Request**. C sends her *PAN* in a digital envelope.
4. **Registration Form**. CA issues a registration form for C.
5. **cardholder Certificate Request**. C fills in the form also with a proposed public key.
6. **cardholder Certificate**. CA issues the certificate for proposed key. Checks?



Goals of SET C.R.

Unicity. If CA stores the keys he certifies, and checks each new key, then no two peers can have the same key certified.

Confidentiality. The cardholder info remain confidential (claim).

Authentication. *Some form* of mutual authentication holds (claim).

SET doesn't support non-repudiation!

Non-repudiation protocols

What is non-repudiation?

A form of *accountability*.

The goal of a non-repudiation service is to collect, maintain, make available, and validate irrefutable evidence regarding the transfer of a message from the originator to the recipient, possibly involving the service of a trusted third party. [Zhou-Gollmann, 1996].

NRO, non-repudiation of origin against the originator . . .

NRR, non-repudiation of receipt against the recipient . . .

Protocol 1

Communication: reliable.

Agents: fair.

1. $A \rightarrow B : nro, B, m, \{\!|nro, B, m|\!\}_{Ka^{-1}}$

2. $B \rightarrow A : nrr, A, \{\!|nrr, A, m|\!\}_{Kb^{-1}}$

Simple!

Protocol 2

Communication: reliable.

Agents: unfair.

1. $A \rightarrow \mathsf{S} : nro, \mathsf{S}, B, m, \{\!|nro, \mathsf{S}, B, m|\!\}_{Ka^{-1}}$

2. $\mathsf{S} \rightarrow B : nrs, A, B, m, \{\!|nrs, A, B, m|\!\}_{K\mathsf{s}^{-1}}$

3. $\mathsf{S} \rightarrow A : nrd, A, B, \{\!|nrd, A, B, m|\!\}_{K\mathsf{s}^{-1}}$

If A initiates, she relies on the server.

Protocol 3

Communication: unreliable.

Agents: fair.

1. $A \to B : nro, B, m, \{|nro, B, m|\}_{Ka^{-1}}$
2. $B \to A : nrr, A, \{|nrr, A, m|\}_{Kb^{-1}}$
3. $A \to B : ack, B, \{|ack, B, m|\}_{Ka^{-1}}$

Message 2 is repeated until ack is received.

Protocol 4

Communication: unreliable.

Agents: unfair.

1. $A \to B : poe, B, \{|m|\}_K, \{|poe, B, \{|m|\}_K|\}_{Ka^{-1}}$
2. $B \to A : acp, A, \{|acp, A, \{|m|\}_K|\}_{Kb^{-1}}$
3. $A \to B : nro, B, K, \{|nro, B, K|\}_{Ka^{-1}}$
4. $B \to A : nrr, A, \{|nrr, A, K|\}_{Kb^{-1}}$

The protocol is unfair on A!

A fair non-repudiation protocol

1. $A \to B : nro, B, L, \underbrace{\{|m|\}_K}_{c}, \underbrace{\{|nro, B, L, c|\}_{Ka^{-1}}}_{NRO}$

2. $B \to A : nrr, A, L, \underbrace{\{|nrr, A, L, c|\}_{Kb^{-1}}}_{NRR}$

3. $A \to \mathsf{S} : nrs, B, L, K, \underbrace{\{|nrs, B, L, K|\}_{Ka^{-1}}}_{NRS}$

4. $B \overset{ftp}{\leftrightarrow} \mathsf{S} :$

5. $A \overset{ftp}{\leftrightarrow} \mathsf{S} :$

$nrd, A, B, L, K, \underbrace{\{|nrd, A, B, L, K|\}_{K\mathsf{s}^{-1}}}_{NRD}$

Resolution of disputes

- B claims having received m from A, but A denies having sent it. B must provide m, c, K, L, NRD, NRO.

  The judge checks that

  1. $NRD = \{|nrd, A, B, L, K|\}_{K\mathsf{s}^{-1}}$
  2. $NRO = \{|nro, B, L, c|\}_{Ka^{-1}}$
  3. $\{|c|\}_K = m$

If checks succeed, then A lies.

Resolution of disputes

- A claims having sent m to B, but B denies having received it. A must provide m, c, K, L, NRD, NRR.

  The judge checks that

  1. $NRD = \{\|nrd, A, B, L, K\|\}_{Ks^{-1}}$
  2. $NRR = \{\|nrr, A, L, c\|\}_{Kb^{-1}}$
  3. $\{\|c\|\}_K = m$

If checks succeed, then B lies.