Secure Electronic Marketplace for Europe

OCG Arbeitskreis "Electronic Commerce"
Outline

- What is SEMPER?
- Reasoning behind design
- Glimpse on the architecture

More information:
- See <http://www.semper.org>
- Final Report, LNCS Springer-Verlag, early 1999
What is SEMPER?

- European R&D Project, 1995-1998
  - ~ 30 people, 20 partners, lead by IBM
  - ~ 13.000.000 CHF
  - EU ACTS Programme: 50% EU & CH, N
- Objectives
  - secure B2B/B2C/P2P e-commerce
  - coherent, complete, extensible framework
  - leitmotif: fairness and multi-party security
- Results:
  - architecture & protocols for secure e-commerce
  - SEMPER electronic commerce agreement
  - prototype, trials & evaluations
  - demonstrator: Fair Internet Trader
Service provision
- Otto Versand
- Eurocom
- Fogra
- Maris

Banking
- Europay
- Commerzbank

Telecom operators
- France Télécom
- Intracom
- KPN Research

Social sciences
- Freiburg Univ.

Security engineering
- Cryptomathic
- CWI
- Digicash
- GMD
- IBM
- Saarbrücken Univ.
- Dortmund Univ.
- SINTEF
Outline

- What is SEMPER?
- Reasoning behind design
- Glimpse on the architecture

More information:
- See <http://www.semper.org>
- Final Report, LNCS Springer-Verlag, early 1999
Current situation

'I want to talk to the secure server'

Certificate really on "Bob"?
Issueing CA trusted? And: so what?

"Now I'm talking to Bob!"
"My messages are secure!"
"If he cheats me I can go to court"

US export-controlled crypto?

SSL is transparent to the application, thus, no signatures!
Current situation

"I want to talk to the secure server"

Secure documents, not (just) secure connections

Certificate really on "Bob"? Issueing CA trusted? And: so what?

US export-controlled crypto?

SSL is transparent to the application, thus, no signatures!

User interface + SECA, Liability Cover Certificates

Cryptography made in Europe
Processes, not just steps

- Processes are the entities that need to be trustworthy, not (just) the single steps.
- Security requirements like anonymity cannot be fulfilled on the step-level.
- Authorization policies naturally depend on processes.
- Small number of generic, evaluated and certified business processes are needed.
Processes are the entities that need to be trustworthy, not (just) the single steps.

Security requirements like \textit{anonymity cannot} be fulfilled on the step-level.

Authorization policies naturally depend on processes.

Small number of \textit{generic, evaluated and certified} business processes are needed.
Dispute handling

Intended run of a payment system: SET, ecash, ...

Does not exist in existing payment systems, e.g., SET!

mi@zurich.ibm.com
Dispute handling

I pay 100 sfr to Bob.

Alice

I pay 100 sfr to Bob

Alice

Deal & transaction browser,

dispute interfaces for payments

Intended run of a payment system:

Does not exist in existing systems, e.g., SET!

mi@zurich.ibm.com
Multi-party security

Centralized trust
- everybody's security is based on trust in security admin

Multi-party security
- make trust assumptions explicit; don't require to put trust in opponents
- minimize trust requirements
- distribute trust on many TPs
- leave some choice whom to trust

e.g., Kerberos
e.g., SET/iKP
Multi-party security

Centralized trust
- everybody's security is based on trust in security admin

Multi-party security
- make trust assumptions explicit; don't require to put trust in opponents
- minimize trust requirements
- distribute trust on many TPs
- leave some choice whom to trust

Symmetric design, all security requirements can be satisfied locally

e.g., Kerberos
e.g., SET/iKP
Trusted computing base

- Virus
- Trojan horse
- Applications with security holes

PC operating systems without much security (and with holes, too)

- Store and use secret signature and encryption keys
- Control what is signed

- Store and use secret signature and encryption keys
- Control what is signed

mi@zurich.ibm.com
Trusted computing base

Create awareness for the problem: "Trusted user interface"

PC operating systems without much security (and with holes, too)
What are the steps?

1. Code loading
2. Negotiation
3. Contract
4. Payment
5. Delivery
6. or
7. Dispute
What are the steps?

Authentication  Confidentiality  Non-repudiation  Fairness

Buyer

Request for offer

Signed Order

Signed Offer

Goods

Receipt

Money

Receipt

Merchant

Authentication  Confidentiality  Non-repudiation  Fairness
What are the steps?

Authentication  
Confidentiality  
Non-repudiation  
Fairness

Architecture based on transfers and fair exchanges

Request for offer  
Signed Offer  
Goods  
Money  
Receipt

Authentication  
Confidentiality  
Non-repudiation  
Fairness

Buyer  
Merchant
Outline

- What is SEMPER?
- Reasoning behind design
- Glimpse on the architecture

More information:
- See <http://www.semper.org>
- Final Report, LNCS Springer-Verlag, early 1999
**SEMPER architecture**

**Business applications**

**Commerce block**
Standard business processes

**Transfers & fair exchanges**
“Containers” + time stamping, contracts, certified mail, etc.

**Payments**
“Money”

**Certificates**
“Credentials”

**Statements**
“Documents”

**Supporting services**
Communication, crypto engine, trusted user I/O (TINGUIN), archive, access control, preferences
Frameworks

Transfers & fair exchanges
- "Containers" + time stamping, contracts, certified mail, etc.

Payments
- "Money"

Certificates
- "Credentials"

Statements
- "Documents"

Supporting services
- Communication, crypto engine, trusted user I/O (TINGUIN), archive, access control, preferences

Payment Mgr

- generic payment api

cheque-like
- Adapter
  - SET

negotiation, compensation, management

MANDATE
- Adapter
  - checks

cash-like
- Adapter
  - e-cash™

Adapter
- Chipper

Smartcard
What is SEMPER?

The Swiss Knife for Secure E-Commerce

- Framework for secure electronic commerce
  - coherent, comprehensive, extensible
- Principles
  - multi-party security
  - linking of secure steps into secure processes
  - document-level security
  - supports secure transfers and fair exchanges