Architecture of SEMPER

Michael Steiner

IBM Zurich Research Laboratory
CH8803 Rüschlikon, Switzerland
Ph. +41 1 724 8286 / Fax +41 1 724 8955
<sti@zurich.ibm.com> / <http://www.zurich.ibm.com/~sti>

Second Public SEMPER Workshop,
November 4, 1998
Electronic Marketplace “Internet”

- Document Exchange
- Mail Order Retailing
- Electronic Publishing
- Ticketing
- Subscriptions
- Information Brokerage
- Auctioning
- ...

Registration & Certification
Directory
Time-stamping
Notary

Internet

Individual

Individual
Retailer
Publisher
Library
Broker
Bank
Government

Bank
Current Situation

- Requirements
  - Integrity
  - Authentication
  - Confidentiality

- Some issues
  - Weak cryptography
  - Trust in CA
  - Trust in Peer
Secure Electronic Commerce

- More than secure communication ...
  - **Multi-party problems**: payments, notarized contract signing, auctioning, copy protection, ...
  - **Multi-party Security**: limit trust in others, make trust explicit, verifiability of trusted parties

- More than payments ...
  - **Processes**: Systems must be securely linked, e.g. contract with payment with delivery

- More than technical security...
  - **Legal and technical foundations**: Digital signatures; registration & certification; secure hardware; unified user interfaces; Dispute handling...
Objectives for Architecture

- **Coherent model as basis**
  - should be easy to understand ..

- **Security as driving factor**
  - security cannot be added later …
  - addresses multi-party security requirements
  - supports dispute handling

- **Openness**
  - Extensibility
  - Uniformity
  - Generality
Scenario: Buying Airline Tickets ...

- Notary Public
- Certification A.
- “Bank,” etc.

Alice

Request → Offer → Reservation Order → Reservation Confirmation → Payment → Payment Receipt → Ticket → Ticket Receipt

BobAir

Third Parties

Dispute Handling
Secure Transfer & Exchanges

- Transfer & Exchanges
  - Data
  - Statements
  - Payments
  - Credentials

- Security Requirements
  - Authentication
  - Confidentiality
  - Non-repudiation
  - Fairness
Bird’s View

Alice

BobAir

Business Application

Certificates
Signed Offer
Signed Order
Fair Contract Signing
Fair Payment for Receipt
Delivery

SEMPER

SEMPER
A closer look ...

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

downloadable
Open Service Block Architecture

- External interface (API)
- Internal interface (SPI)

Manager for X

Module for Service X, Type 1
  - Adapter
  - Implementation

Module for Service X, Type n-1
  - Adapter
  - Implementation

Module for Service X, Type n
  - Adapter
  - Implementation

etc.
Accountability: Implications for Design

- Certificate cannot bind a user *per se*
- Ideal registration:
  - CA and user sign a contract
  - Certification policy
  - Accepted liabilities
  - Contents of certificate
- User interface
  - Standard GUIs
  - Standard presentation SW
  - “Point of no return”
- Secure key handling and signature generation
  - secrets generated by user
  - secrets never leave trusted user device (e.g., electronic wallet)
- Notarization of signature

Did supposed signer accept liability for such signatures?

Was supposed signer aware of signed contents?

Was supposed signer the creator of a signature?
Commerce Layer

◆ Deals
  ◆ business context with negotiated quality of service
  ◆ secure linkage of commerce transactions
  ◆ collection of evidence, deal browser & dispute handler.

◆ Commerce transactions
  ◆ Extensible class hierarchy of primitive transactions (payments, offers, orders, ..)
  ◆ Core classes enforce standardized presentation of information to the user and proper authorization

◆ Downloadable extensions
  ◆ certification
**Trusted Interactive Graphical User Interface**

- **Why?**
  - Never trust a window on an untrusted PC or in a browser ...

- **How?**
  - Approximation in software: Dedicated window
  - Ideal solution: “Electronic wallets” with keypad & display

- **More general problem: Untrusted Hardware**
Need for Trusted User Devices

- Software-only
  - Secret key + computation + protocols
  - User

- Smartcard-only
  - SMART CARD
  - Secret key + computation
  - Protocols
  - User

- User device + security module
  - Secret key + computation + protocols
  - User
Electronic Wallet: Design Example

Plugin-in Security Modules

Infrared
Display
Keypad

Back
Front

© N. Özalp, Fachhochschule Hildesheim/Holzminden/Göttingen, 1995
Conclusions

Sound architecture, solid bricks

Supports multiple business models:
• business-to-business, business-to-consumer, private-to-private
• symmetric design

Comprehensive:
• multi-party security at its core
• processes, not just steps
• large set of supported services

Extensible:
• service framework
• generic interfaces
• downloading of modules, BAs

• SECA
• liability cover
• deal concept
• fair exchanges
• signed offers/orders
• trusted user interface
• pervasive anonymity
• dispute handling