Universidad Carlos III de Madrid

University Carlos III of Madrid

Module 1: Traditional E-commerce

Secure E-commerce
OpenCourseWare

José María de Fuentes, A.I. González-Tablas, A. Ribagorda

Main objective

The development of a traditional merchant environment.

Main Topics

- Development of an merchant environment that corresponds to the schema used in "traditional" merchant.
- Get to know the role of each one of the entities participating in a merchant transaction, their needs, security issues and possible attacks launched by those entities.
- Analyze security issues that affect this schema, choose tools for solving them and observe the effect of applying those solutions to the schema.

Technologies to be used

- Java
- XML
- SSL
- Communication through Sockets

Module's main tasks

To successfully pass this module, the following tasks are needed:

- **Analyzing the security requirements for each entity** involved in the merchant process proposed in this first module. As a result of this analysis, some improvements to the schema shown in this document should be proposed (e.g: message encryption, etc.).

- **Implementing the proposed architecture, including the security enhancements proposed in the previous analysis.**

Environment description (see figure 1)

The consumer has navigated the Internet and has already made a selection of required items. It then confirms the selected items, quantity and final price to be paid in an operation called **CHECKOUT**.

Right after this operation, the customer's data for authorizing the payment is transferred to the merchant (step 1). The merchant creates an "Authorization Request" to be sent to the merchant's Bank (step 2). The merchant's Bank processes the authorization and sends it to the Consumer's Bank, which evaluates the request and sends the appropriate response to the merchant's Bank (step 3). Now, the merchant's Bank generates and sends the corresponding response for the merchant (step 4). Finally, the merchant sends its response to the consumer (step 5).

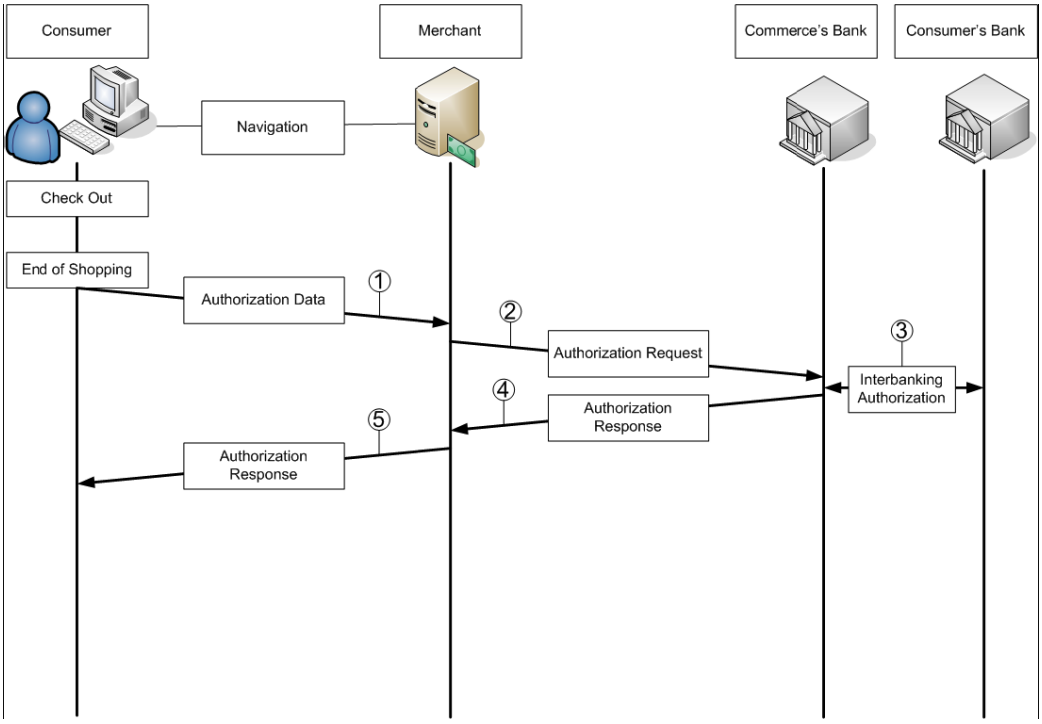

Figure 1.

Entity description

Consumer

The entity "consumer" includes the user itself and the application used to carry out the purchase operation. Their characteristics are:

User:

• Has a credit card issued by the "consumer's Bank", with a **PAN** (Personal Account Number), the first name and last name and the expiration date. It also has a **CVV2** code (three number code that appears under or above the signature area, on the back side of the card) and the signature.

•The user has selected the item to be bought, confirmed the final price and now it has to enter its personal data (including those on the credit card) to proceed with the final phase of the e-shopping.

***Application*:**

• The Application is like a reduced browser that the consumer uses in order to buy. This "browser" only sends and receives the necessary messages to proceed with the shopping. Even though HTTP-like messages have to be used, NO implementation of the HTTP protocol is necessary.

• In case of an error, the Application has to show an error message and the shopping is aborted.

• The Application is the only means for the consumer to send and receive information about the ongoing of the process.

Merchant

• The merchant entity includes all services needed to support an e-commerce operation (databases, web servers, application servers, etc.) needed to proceed with the purchase and payment transactions, as well as additional software and other required resources. Not all of these services will have to be implemented in this work.[1]

• Initially, the merchant loads any necessary data from XML files, and listens for incoming consumer connections. When a connection is established, the merchant will await incoming data for sending the Authentication Request to its Bank.

• Right after this Request, the merchant waits for the Response that communicates if the operation is or not authorized. When this Response arrives, the merchant sends the corresponding Response to the consumer.

Merchant's Bank

• The merchant's Bank is responsible for receiving Authorization Requests from the associated merchants, and contacting the corresponding consumer's Bank to process the Requests.

• As soon as the merchant's Bank obtains the Response for a Request it will send it to the merchant that requested the authorization.

Consumer's Bank

• The consumer's Bank participates in the Interbanking Authorization process by receiving an Authorization Request from the merchant's Bank and validating it. In case that this Request can be successfully processed, necessary operations (such as updating accounts' balances) must be performed. After

these operations have been completed, the consumer's Bank sends the Response to the merchant's Bank with the results of the transaction enclosed.

Proposed message content

- Authorization Data
  - PAN
  - Expiration Date
  - CVV2
  - Name
  - Last Name
- Authorization Request
  - Operation ID
  - PAN
  - Expiration Date
  - CVV2
  - Name
  - Last Name
  - Total Amount
- Interbanking Authorization
  - Operation ID
  - PAN
  - Expiration Date
  - CVV2
  - Name
  - Last Name
  - Total Amount
- Interbanking Response
  - Operation ID
  - Authorization Status
  - Total Amount
- Authorization Response
  - Operation ID
  - Authorization Status
  - Total Amount

Whenever an error occurs, additional fields for the error code and error messages may be added if needed, in order to inform the entity which received the message about that error. Any other changes and enhancements to this proposal must be reasoned and justified.

Requirements & Restrictions

• The implementation has to be developed in **Java**, and it must work in computer laboratories at University. Libraries not included in the standard distribution of the current JDK are not allowed (at least for this module).

• **Every message** sent from one entity to another has to be formatted **as an XML document**.

These XML documents must follow the restrictions given by the Internet Engineering Task Force (IETF) for XML version 1.1, and **must be validated against the corresponding DTDs** (to be created by the students). In order to process XML documents within Java programs, SAX libraries can be used.

• **Students are not allowed to introduce new messages** in the proposed schema.

• Security improvements to be introduced must be **justified and feasible** (that is, their implementation in a real merchant scenario should be possible).

• In order to avoid conflicts and problems, the following communication ports have to be used for plaintext communications: **8000, 8008, 8080, 8800**, etc.

• Similarly, communication ports to be used for secure communications are: **4433, 4443, 4333, etc.**

• There should not be any difference in the implementation of the Consumer's Bank and the Merchant's Bank: a *Bank* must be able to act as a Consumer's Bank or a Merchant's Bank. In order to identify Banks, a four digit code will be used, and any credit card issued by a Bank must have the same first four numbers of the PAN as the Bank ID.

It is recommended to start with the source code and the examples provided by the teachers.