IELM 231 IT for Logistics and Manufacturing
Client-side programming - part III

Background:
Up until now, we have seen how a web-user can submit data to a server, and the server can then execute CGI programs to take some actions based on the inputs. Since ***CGI programs*** run on the web server, they ***are caller server-side programs***.

In this sequence of ***labs***, we will see how to enhance client-server program functionality by also asking a program to run on the client side via the web interface. Namely you will write ***client-side programs***.

The main operation is as follows:
1. Web client (e.g. browser) sends a request for a web-page (URL)
2. Server sends the requested page, which contains data+program(s)
3. Client executes the program, which may (i) do something, (ii) generate some more data to display; then the client displays the resulting data as an HTML page.

In the following labs, we shall see what type of things can be done by client-side programs, and in particular, use them for some common and useful client-server tasks. In particular, you create simple applications for ***the following three tasks***:

Task 1. Simple client-side program to ***create a better GUI***, including checking for proper inputs, verification of data, etc. This exercise will also include basics of ***Javascript*** programming.

Task 2. ***Sessions tracking*** using cookies.

Task 3. A simple ***secure communication*** exercise, using ideas of ***symmetric keys***.

In this exercise, we look at task 3.

Objectives:
(a) Continue learning basics of Javascript language.
(b) Learn a simple and efficient algorithm for encryption (we shall use the *Tiny Encryption Algorithm*, or TEA, which is based on the Feistel cipher)
(c) We shall assume that both, the client and server know the secret password, or key. The TEA algorithm has to be programmed separately for client and server sides. On the client side, encryption will take place by Javascript, which is embedded in an HTML form requested by the client; when the form is submitted, the data is encrypted before being sent. On the server side, a CGI program (you may use VB 6.0) will retrieve the key from a Database, and use it to decipher the transmitted data.

Source code for the TEA cipher is available in several different programming languages, including Javascript. You can read about how TEA works, and also obtain the code from the website linked below.

Step 1.We shall use the same table, *ClientData*, as the one you used in task 1 of this lab. However, we shall add one more field to the table: password char(20).
Table name: ClientData
Fields:
LastName char(30),
OtherNames char(50),
CCno char(16),
CCtype char(15) which should be one of [“Master Card”, “Visa”, “American Express”, “Diners Club”]
ExpiryMMYY char(5) which should be of the form “mm/yy” (= month/year)
Password char(20)

Step 2. Again, we shall modify the HTML file, SendCCData.html that you wrote for the lab in Task 1.

(i) First add one more text input to the Form, called password. Please make this field as type=password, so that when the user types the password, it does not appear on the screen.

(ii) Modify the TEA javascript code, and add one more function to the event OnSubmit (which occurs when the ‘submit’ button is clicked by the user. As before (completed in Task 1), all inputs are first verified to be in correct format. After that, the fields: CCno and CCType are encrypted using TEA. The encrypted data is sent to the server.

Step 3. Write the CGI program to handle the inputs received from the Form in Step 2. The CGI script first connects to the DB, and retrieves the password of the user. To simplify the programming effort, we will assume that this user already has a record in the DB. The CGI then executes the following tasks:

(i) It gets all the data sent by the form.
(ii) It uses the password from the DB, and de-crypts the data for the CCNo and CCType fields. To do so, you will need to program the TEA cipher algorithm in VB 6.0. This is the main task of this lab.
(iii) It then verifies whether the CCNo and CCType of the user matches with the one in the DB. If yes, the CGI sends a message back to the browser: “Thank you, your data matches our records”. If not, then the CGI sends a message to the browser: “Sorry, data mismatch. Please use the browser back button and try again”.

For learning basic commands of JavaScript, and how to use it in an HTML page, you are encouraged to use any one of the tutorials available online, e.g. the Webmonkey tutorial (lesson 2 covers cookies):
http://www.webmonkey.com/webmonkey/programming/javascript/tutorials/tutorial1.html

How TEA works and source code: http://143.53.36.235:8080/tea.htm

---