

SECURITY OPTIMIZATION OF A DISTRIBUTED APPLICATION FOR CALCULATING DAILY CALORIES CONSUMPTION

Mihai Doinea¹

Abstract

General concepts of calorie expenditure are presented, revealing the needs and the main factors that influence the overall calculation of calorie need for the human body. A distributed application for helping individuals manage daily calorie consumption is presented and ways of optimizing the daily calorie need is achieved by offering different daily menus depending on the intensity effort calculated a priori. Security optimization criteria are presented for distributed applications and an open source encryption algorithm is used for implementing one of the optimization criteria discussed.

Keywords: distributed applications, security optimization, calories, daily menus.

1. Introduction

Calories are the main fuel of human machine. Since the society has become more and more commode, the lack of physical exercises started its way up, revealing different important problems in human's health:

- obesity problems;
- cardio – circulatory problems;
- diabetes;
- high cholesterol levels.

The calorie consumption has become changing the balance in favour of the calorie input process rather than the calorie output process. This means that more are the calories that are entering inside our metabolism than the ones that are leaving by means of physical exercises. This anomaly is the cause of all our metabolism dysfunctions, ultimately leading to different types of illnesses.

The fact that the human body, like all others systems, is under commitment of the first law of thermodynamics, the conservation of energy which results from that, causes various effects depending on the actions done upon the energy, as calories, that is entering into the system.

The main factors that could influence the daily calorie consumption in the human body are:

- physical effort;
- emotional effort;

¹ Mihai Doinea, Amsterdam University, Information Management, The Netherlands, email: l.m.doinea@student.uva.nl

- neurological effort.

The total intake on daily basis is dependent upon the following factors for each individual: age, sex, weight and height.

In [1] statistics with the daily calorie need are presented based on two input factors:

- weight;
- effort intensity: sedentary, low active, moderate active, active, very active, super active.

Table 1 reveals the calorie expenditure for men in a range of weight between 100 and 200 pounds per individual.

Table 1: Man daily calorie need [1]

Men Weight	Super Active	Very Active	Active	Moderate Active	Low Active	Sedentary
100	2600	2350	2100	1850	1600	1350
110	2800	2530	2260	1990	1720	1450
120	3000	2710	2420	2130	1840	1550
130	3200	2890	2580	2270	1960	1650
140	3400	3070	2740	2410	2080	1750
150	3600	3250	2900	2550	2200	1850
160	3800	3430	3060	2690	2320	1950
170	4000	3610	3220	2830	2440	2050
180	4200	3790	3380	2970	2560	2150
190	4400	3970	3540	3110	2680	2250
200	4600	4150	3700	3250	2800	2350

Comparable values are presented also for females, in table 2, but with minor differences between the total figures.

Table 2: Women daily calorie need [1]

Women Weight	Super Active	Very Active	Active	Moderate Active	Low Active	Sedentary
80	1600	1440	1280	1120	960	800
90	1800	1620	1440	1260	1080	900
100	2000	1800	1600	1400	1200	1000
110	2200	1980	1760	1540	1320	1100
120	2400	2160	1920	1680	1440	1200
130	2600	2340	2080	1820	1560	1300
140	2800	2520	2240	1960	1680	1400
150	3000	2700	2400	2100	1800	1500
160	3200	2880	2560	2240	1920	1600
170	3400	3060	2720	2380	2040	1700
180	3600	3240	2880	2520	2160	1800

The fact that women are in need of less calories than men is due to the metabolic rate which is also lower at women than men.

2. Distributed Application

The application for daily calorie consumption is meant to give help to individuals who want regularly check their calorie schedule, building them menus based on the intensity effort, daily set it.

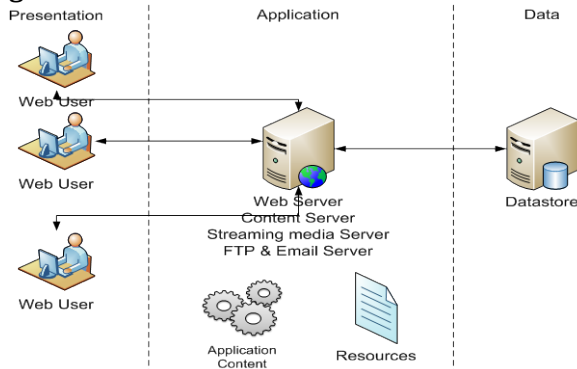
The **Secure Distributed Application for Calories Optimization**, SDACO is built on several layers having a three-tier distributed architecture [2]. The three layers are represented by:

- presentation layer- is given by the interface of the application designed to help users resolve their needs;

- database layer - is represented by a Oracle Database Scheme which stores all the application data;
- application layer - contains all the distributed application main code composed of C# classes;

The application's architecture having the three layers mentioned above is presented in figure 1.

Figure 1: The Three-Tier Web Based Distributed Application [2]

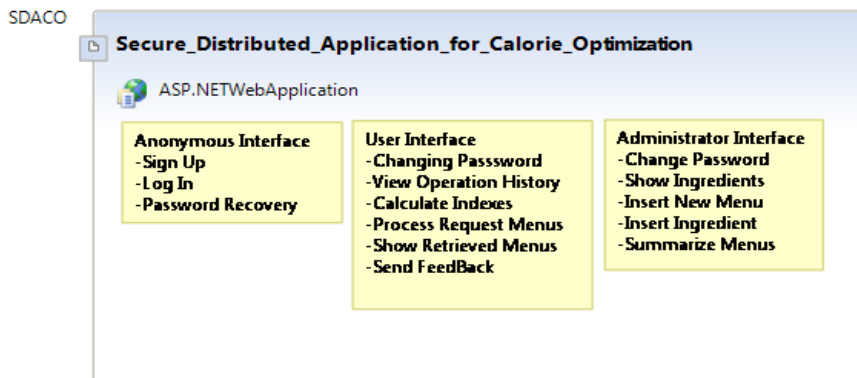


The presentation layer is described by the user's interface which is structured in three major activity zones:

- anonymous functions;
- user's functionality;
- administrator's tools.

Users can attend the following functions in a one way informational flow with the possibility to come back in different region of the process as presented in the figure 2.

Figure 2: Presentation Layer for SDACO



The application layer is defined by the totality of the server-side and client-side source code which intermediates between user's actions and the database layer. The application layer is written using different programming and description languages which are cooperating in achieving the higher level of efficiency and performance, such as:

- C Sharp, C# implemented in the .Net Framework as a managed code language running under a CLR, Common Language Runtime virtual machine;

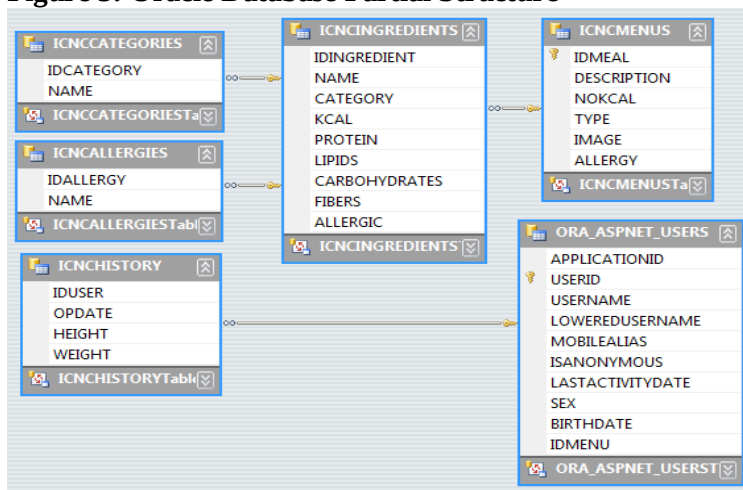
- AJAX for efficiently achieve all the asynchronous procedures defined in the application; AJAX stands for Asynchronous JavaScript and XML;
- JavaScript for successfully implementing client-side functions;
- HTML, a markup language over which ASPX is implemented;
- CSS, Cascading Style Sheets for managing the application's layout.

The database layer contains two categories of information:

- user related information like accounts, roles and memberships along with a history table of each operation undertaken in application;
- menu related information like ingredients, menus, allergies, categories.

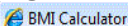
The database structure is defined in figure 3:

Figure 3: Oracle Database Partial Structure



Based on these three layers, the application is offering a diverse menu for each user based on their particularly needs and characteristics. The first step of the application is the Indexes Calculator, based on several inputs like weight (Kg) and height (cm) along with some fixed characteristics as age and sex provided when the user account is created. In figure 4 is presented the interface which uses all the formulas from [1] to provide the end results.

Figure 4: Indexes Calculator

 BMI Calculator

Application for daily calorie expenditure

Welcome mihai

These are your last values from the configuration on 4/9/2010 9:31:33 PM

Please enter your details to calculate your new values:

Weight(Kg)

Height(cm)

Body Mass Index - BMI

Basal Metabolic Rate - BMR

Lean Body Mass - LBM

OPTIONS

[Logout account](#)

▼ User Menu

▷ Change Password

▷ View History

Copyright 2010 - Academy of Economic Studies
Bucharest, Romania

After the indexes were calculated the next step in application is to calculate the total daily need of calories based on the intensity effort defined for each day of the week as showed in figure 5.

Figure 5: Daily Calorie Calculator

Menu Configuration

Choose your allergy:

☐Crustacee☐Alune☐Nuci☐Peste☐Oua

Days	Effort	Colories Needed	Found Menu	Difference	Printing Option
Monday	No Effort	1999.2	2000	0.8	☐
Tuesday	Less Active	2290.75	2300	9.25	☐
Wendsday	Moderate	2582.3	2580	-2.3	☐
Thursday	Strong	2873.85	2880	6.15	☐
Friday	Exortion	3165.4	3154	-11.4	☐
Saturday	No Effort	1999.2	1990	-9.2	☐
Sunday	No Effort	1999.2	1997	-2.2	☐

In this point of application presented in figure 5, based on the five value intensity effort option, the daily calorie need is calculated along with the menus found for every day.

More, the users can filter their menu choosing from a list of allergies which can cause them anaphylactic shocks.

On the next stage a detail version of each menu can be viewed by accessing the link over the found menu value, as presented in figure 6 where the menu found totals 3154 kcal from a 3165.4 calories needed that day.

Figure 6: Menu Detail for Friday

Day: Friday			
	Description	No KCal	Photo
Breakfast	Mio Dejun		
	Cafea filtru : 200 ml Zahar tos : 25 g Unt : 30 g Branza topita : 25 g	1350	
Lunch	Salata calda cu bacon afumat		
	Ingrediente	198	
	2 cartofi noi o salata		
	Piept de pui cu rosii		
Lunch	Ingrediente	1200	
	2 x 500 gr rosii in bulion 6-8 catei usturoi 500 gr piept de pui		
	Mousse au chocolat		
	Ingrediente	50	
Dinner	50 g ciocolata amara 40 g lapte 1 galbenus		
	Paste cu crab		
	Ingrediente	257	
	seminte de fenicul 1 ardei lute oaja si zeama de lamaie		
Dinner	Frajitura cu ciocolata		
	Ingrediente	120	
	100 unt 100 g ciocolata amara pudra de cacao		

These are the options that an individual can access from this application. The whole part of menu management, meaning inserting new ingredients and menus is done through the administrator account interface.

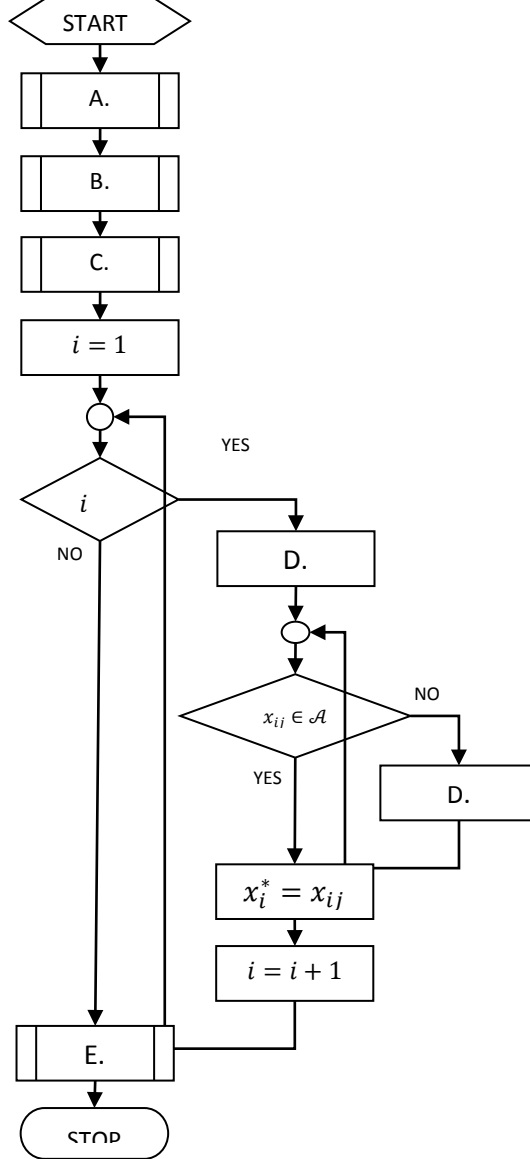
3. Application algorithm

The process of calculating the body indexes for the application along with the daily caloric level is presented in [1], revealing all the correlations that are part of this system of equations. The algorithm for compounding the menus to meet the daily calorie level has the following steps:

- calculating input data – daily calorie intake based on the effort intensity level;
- creating the vector off allergies to meet the individual requirements;
- choosing the menus from the database according with the allergy vector for each one of the menu types: Breakfast, Lunch First Meal, Lunch Second Meal, Lunch Dessert, Dinner Meal, Dinner Dessert by creating different arrays, $A_k, k = \overline{1,6}$ part of $X = A_1 \times A_2 \times A_3 \times A_4 \times A_5 \times A_6$ – the set of all solutions;
- from X only the solutions that are allowable are chosen for each day of the interval; an allowable solution x_{ij} is part of the set $\mathcal{A} \subseteq X$ defined as $\mathcal{A} =$

- $\{x_{ij} | x_{ij} \in X, \|x_{ij} - TDI_i\| < \varepsilon\}, i = \overline{1, n}, j = \overline{1, Card(\mathcal{A})}$ and $\varepsilon = 0.005 * TDI_i$;
 n – total number of days of the evaluation period; TDI_i – total daily intake (kcal);
 E. final menus are assembled and the results from figure 6 are presented.
 The logical schema of the presented algorithm is depicted in the following figure

Figure 7: Algorithm Logical Schema



For being more efficient and less time consuming due to the massive volume of menu combinations given by the Cartesian product off all menu sets, the distributed application can provide filtered sets of menus for the algorithm based on the allergy filters that a user can activate. As a solution is validated as allowable, the process is started again for other sets of values.

4. Security Optimization Algorithm

Optimization in general refers to a process of improvement by transforming A into B with loss of resources, but performance gain which helps the entire system to be more efficient.

Optimization:

- uses existing resources;
 - achieves better quality measured by specialised built-in metrics;
- The optimization problem is defined as having two components:
- input – all the information that is given about the problem which need to be solved;
 - output – the sought element which is resulting as part of the optimization process.
- The given part is described by:
- *optimization function* $f(x) = y$ defined as $f: A \rightarrow B$;
 - *constraints* over the $x \in A$ and $f(x) \in B$.
- The sought part is described by:
- $x_0 \in A$ for which $f(x_0) \leq f(x)$ referring to minimization or $f(x_0) \geq f(x)$, maximization;
 - $x^* \in A$ defined as a local minimum for which there exists some $\delta > 0$, so that for $\forall x$ such that $\|x - x^*\| \leq \delta$, the expression $f(x^*) \leq f(x)$ holds;
 - $x^* \in A$ defined as a local maximum for which there exists some $\delta > 0$, so that for $\forall x$ such that $\|x - x^*\| \leq \delta$, the expression $f(x^*) \geq f(x)$ holds.

For optimizing security in a distributed application, we have to take into account the following possible optimization criteria:

- increasing the general level of security until the point where any additional security increase is not worthy of its investments;
- decreasing the degree in which vulnerable information are partially or totally revealed to end-users;
- eliminating vulnerabilities that concord to specific risks evaluated using the risk evaluation diagram presented in [2];
- increasing the efficiency of a security component, part of the security system of a distributed application;
- minimizing the total amount of time for processing secure transactions between application components.

Following we will discuss the matters implied by the degree of vulnerable information criteria, giving example on the SDACO implementation.

In the process of communication between different parts of application, important and sometimes vulnerable information is vehiculated. The process of communication, must be, in this sense, well controlled, taking care that there's no possibility to gain control over the resources using sensitive data skipped to end-users in the communication process.

Some of the ways of accessing data from the communication process between application components is by checking the following objects for any sensitive material:

- URL's – the internet addresses used to navigate through the application; sensitive information can be revealed in this region as part of the web submission process of a form through the GET method [3];
- Cookies – temporally client-side files used for storing different setting data for the web based distributed application [4].

In the process of transferring information about the daily menus for showing the results from figure 6, the following syntax, $S1$, for the URL is used by the application: $S1 = \text{http://www.webserver/pathASPXfile?param1=value1\¶m2=value2\&..\¶m7=value7}$, where:

- webserver – icnc.ase.ro;
- pathASPXfile – application file path of the requested resource;
- $param_i, i = \overline{1,7}$ - the parameters' names for sending menu ID's for the next resource;
- $value_i, i = \overline{1,7}$ – the parameters' values consisting in menu ID's and day ID.

Having these types of information at its disposal, a user can actually see throughout the whole sets of menus, just by modifying the parameters values and submitting again a valid request to the web server.

In this sense, an optimization in terms of minimizing the value of the indicator **Degree of Sensitive Data**, DSD is required:

$$DSD = 1 - \frac{TSDP(Kb)}{TSD(Kb)},$$

where:

TSDP – amount of sensitive data which is protected by user's access;

TSD – amount of total sensitive data vehiculated by the application.

The DSD indicator takes values between $[0; 1)$ in the following scenarios:

- $DSD = 0$, when all sensitive information vehiculated in the application is protected;
- $\lim_{TSDP \rightarrow 0} DSD = 1$ because $\lim_{TSDP \rightarrow 0} \frac{TSDP}{TSD} = 0$ and $DSD = 1 - 0 = 1$.

An implementation of this optimization process was conducted in the application by encrypting the URL syntax presented above with an open source encryption algorithm from **One Time Pad**, OTP class algorithms which revealed itself to be the most efficient from a list of five open source algorithms presented in the paper [5].

The security optimization process takes place under the following step procedure:

- identifying the parameters' values and creating the $S1$ URL syntax;
- encrypting $S1$ into $S2$ by means of the encryption function $OTP(S1) = S2$;
- creating the new URL, $S3$ from $S2$, based on the URL Base64 Encoding representation for being able to be sent over the internet;
- the host page gets the URL $S3$ syntax from the request object of the page and decodes it from the Base64 representation, transforming it into $S2$ encrypted state;
- from the encrypted $S2$ state, the mirror encryption operation takes place, as the OTP are symmetric class algorithms, and the final $S1$ syntax is decoded;

- the S1 syntax will serve to successfully retrieve all the menu ID's and day ID for showing the results from figure 6.

The final URL syntax which travels the network and is transparent to the user is like $S3 = \text{http://www.icnc.ase.ro/User/MenuDetail.aspx?pass=ARaQQtorYdYiL7wHrQusDs4Do8WrtQ2}$.

The following table shows an improvement of the DSD indicator due to growth of the $TSDP$ value.

Table 3: DSD Optimization Chart

Without Security	Relation	With Security
$TSDP_0(Kb)$	$<$	$TSDP_1(Kb)$
$\frac{TSDP_0(Kb)}{TSD(Kb)}$	$<$	$\frac{TSDP_1(Kb)}{TSD(Kb)}$
$1 - \frac{TSDP_0(Kb)}{TSD(Kb)}$	$>$	$1 - \frac{TSDP_1(Kb)}{TSD(Kb)}$

From table 3 is resulting $DSD_1 < DSD_0$, meaning an optimization of the DSD as it was defined.

5. Conclusions

In a society where information may be the key which makes the difference between profit and non-profit, this kind of approach is meant to rein organizations, who are implementing web-based distributed applications, to achieve their goals.

By pursuing the security optimization, organizations must take into account not only the technological factors, but as well the non-technological ones, like human resources and ensure a correct and efficient security policy implementation.

Acknowledgements

This article is a result of the project POSDRU/6/1.5/S/11 „Doctoral Program and PhD Students in the education research and innovation triangle”. This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies, project no. 7832, Doctoral Program and PhD Students in the education research and innovation triangle, DOC-ECI.

Bibliography

- [1] Mihai DOINEA – *Distributed application for calories optimization*, Body Building Journal, vol. 1, no. 1, 2009, pp. 72 – 83, ISSN 2066-8007
- [2] Catalin BOJA, Mihai DOINEA – *Security assessment of a web distributed application*, Informatica Economica Journal, vol. 14, No.1, 2010, pp. 152 – 162, ISSN 1453-1305
- [3] Word Wide Web Consortium, W3C – *Hypertext Transfer Protocol – RFC*

- 2616, available at: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [4] Word Wide Web Consortium, W3C – *HTTP State Management Mechanism – RFC 2109*, available at: <http://www.w3.org/Protocols/rfc2109/rfc2109>
- [5] Mihai DOINEA – *Comparative analysis of performance encryption and decryption open source procedures using OTP cryptographic class algorithms*, Open Source Science Journal, vol. 1, no.2, 2009, pg. 31-46, ISSN 2066 – 740X