

**Paper for the *Educators and Planners: Symphony or Discord* Conference  
AAIR Conference 1 – 3 December 1999**

**PROTOTYPE WEB-BASED ON-LINE ENROLMENT SYSTEM:  
A CASE STUDY ILLUSTRATING ISSUES OF  
DATA VALIDATION AND SECURITY**

**Soroush SEDAGHAT**  
Management Information Analyst  
University of Western Sydney Macarthur

**Josef PIEPRZYK**  
Associate Professor  
Centre for Computer Security  
University of Wollongong

**Steve HANSEN**  
Senior Lecturer  
Faculty of Informatics, Science and Technology  
University of Western Sydney Macarthur

**Yogesh DESHPANDE**  
Senior Lecturer  
Faculty of Informatics, Science and Technology  
University of Western Sydney Macarthur

**ABSTRACT**

Universities in Australia are requested by the Federal Government to provide a variety of information about enrolment details and students each year. Associated with this are many data errors detected by the government-supplied software, DETYAPAC, and also the major concerns for the accuracy, integrity and consistency of data collected at enrolment time. To overcome data errors, by accurately and electronically capturing and validating student statistics in real-time situations, a Web based On-line Enrolment System has been initiated and prototyped at the University of Western Sydney in Australia.

In this paper we introduce the On-line Enrolment system, the techniques for assisting students to enter correct information in electronic forms (e-forms) and the methods by which students interact with the system. We step through various aspects of the prototyped data acquisition system and describe the rationale behind the system. Then we critically analyse different data types, system security and privacy matters and suggest solutions to minimise the risks and associated Web security problems. We also suggest the need for establishing a National Certificate Authority for Australian universities in order to issue, maintain and oversee the Public Key certificates required for secure communication. We comprehensively examine verification of electronic form originality by different methods and discuss their related problems. We suggest the *Form Validation Protocol (FVP)*, using cryptographic techniques, to effectively prevent submission of false and incorrect information by a perverse user through fabricating an artificial form to the system database.

# PROTOTYPE WEB-BASED ON-LINE ENROLMENT SYSTEM: A CASE STUDY ILLUSTRATING ISSUES OF DATA VALIDATION AND SECURITY

## 1. INTRODUCTION

The accuracy, integrity and consistency of student data collected at enrolment time has been an issue with the government department that provides the funding to the institutions and the recipient universities. Together with other data collections, the Government also uses the student data to provide further access to target equity groups and also monitors higher education performance through quality assurance procedure [4], all of which require sound data sources. There are two kinds of data contamination. The first kind relates to the mistakes committed by students when filling out conventional forms. The second kind occurs when the collected information is being incorrectly entered into the Student Records Systems (SRS). Currently used traditional paper based systems are not user friendly, and do not provide user awareness of incorrectly completed or missing information. They also are not automatically linked to the SRS. Moreover, these forms are to be entered into the SRS via the data entry procedure, which not only adds to the cost of the concerned units, but also brings forth a potential source of further errors. As have been detected by the government supplied software, DETYAPAC, these errors affect the quality and integrity of data, which is the prime information source used by government agencies for funding purposes, decision-making, university planning and similar applications.

The recent advancements in the Internet and World Wide Web (WWW) technology enable us to step beyond the original ways of collecting and reporting information and to develop and deploy Web-based on-line registration systems. WWW has significantly evolved from its original purpose of publishing information only, into an interactive platform of concurrent engineering including video, sound, animation and dynamic links to other sites [8]. Web Engineering [10] has significantly changed the way that Distributed Computing Systems were supposed to work, by allowing mobile code and applets to travel through the Internet and run on client's machine [7]. This has brought forth many new applications such as *incorporating data validation and redundancy checking into electronic forms*. Validation modules, supported by programming languages such as JavaScript, assist to correctly capture the intended information, based on certain business rules, on-line in real-time situations. Such systems are user friendly and ideally suited for users with no or limited computer skills. Typically, such systems are platform independent and do not require any special hardware and/or software apart from the minimum Web browsing facilities. They are cost effective when compared to the traditional student registration systems.

To minimise, or possibly eliminate, problems with the correctness and integrity of data, the first author has initiated a Web based On-line Enrolment System and prototyped at UWS Macarthur. This system interactively and progressively records student enrolment details including personal details (as per government specifications), the details of the Higher Education Contribution Scheme (HECS) and course and subject selection particulars. In order to validate student information, client-side JavaScript programs carefully examine the student input. Additionally, certain cross validation routines check the relationship between one piece of the information and the other before the form is submitted to the server. The rules, governing such validations, have been programmed based on the criteria specified by DETYA [2,3].

The Active Server Pages (ASP) environment has been employed for establishing a dynamic interface with the users of the system. The site has been set up using NT4.0 operating system and Microsoft Internet Information Server (IIS). The site has been configured for commencing (new) and continuing students who are recognised by the system as valid users, i.e. students who have been admitted through the Universities Admissions Centre (UAC) or registered via Direct Entry Mode or other admission schemes. Initially the student database has been populated with the student information obtained from the UAC data files (for new students) or from previous study records in SRS (for

continuing students). A student with a pre-assigned username and password registered in the site's databases, is a legitimate user and hence can obtain access to the site.

There are certain security issues that need to be addressed in the development of our On-line Enrolment System. We discuss the two following security problems: the validation of data by a user (student) and the authentication of electronic forms sent from browsers to the front server of the SRS.

The paper is structured as follows. Section 2 describes methods and techniques used to verify the correctness and integrity of data supplied by users. In Section 3, our on-line data acquisition system is presented. Security and privacy aspects of the system are studied in Section 4. An authentication protocol used in the system is proposed in Section 5. Section 6 concludes the paper.

## 2. TECHNIQUES FOR ENFORCING CORRECT DATA ENTRY IN E-FORMS

Electronic forms (e-forms) can be seen as collections of components comprising of push buttons, text boxes, select lists, check boxes, radio buttons, drop down menus and a series of hyperlinks (to enable the creation of information nodes) in conjunction with relational databases, which contain student records. Table 1, describes the logic and techniques associated with each Form components.

**Table 1 - Form components and related techniques**

<b>Form Components and Techniques</b>	<b>Description and applications</b>
<ul style="list-style-type: none"> <li>• <b>Drop down list</b></li> </ul>	This component allows the user to provide only one answer from the list of available options. It prevents typing errors. In the prototyped enrolment form this has been used for fields such as Date of Birth, Language Spoken at Home, Country of Origin, Post Codes and fields of similar nature (see Figure 1).
<ul style="list-style-type: none"> <li>• <b>Radio Button</b></li> </ul>	This facility enforces users to submit one answer only. In the context of this prototype, it has been applied in the area of Prior Education Studies, Gender, Aboriginality, Disability, Citizenship and in all fields, where a single answer is required.
<ul style="list-style-type: none"> <li>• <b>Checkbox</b></li> </ul>	This ensures that users select a valid collection of answers. For example, types of disabilities from a list of six options (see Figure 2).
<ul style="list-style-type: none"> <li>• <b>Mixture of Radio Buttons and Prompts</b></li> </ul>	Prompts normally guide users to correctly fill out forms if they enter unexpected or erroneous information. For instance, turning on a radio button indicating usage of a language other than English at home but not selecting the name of the language. This information is considered to be incomplete and a prompt will be sent to the user.
<ul style="list-style-type: none"> <li>• <b>Combination of Radio Buttons and Select Box</b></li> </ul>	The option is used to obtain additional information when a partial answer has been supplied. When a radio button is selected, then the drop down list(s) associated with other radio button(s) are turned off, guiding the user to select an answer from the available select box. This technique is used in questions related to Prior Education study and the completion years associated with each section.
<ul style="list-style-type: none"> <li>• <b>Combination of Checkboxes and</b></li> </ul>	The technique associates the functionality of certain radio buttons with some checkboxes. For example, the relationship between the

### Radio buttons

disability and its type. Figure 2 illustrates the use of checkboxes in the Personal Details Form where students are asked to enter information regarding disability. If they answer "yes" to the question asking for their disability, then the checkboxes are activated otherwise they stay inactive. This prevents recording of inconsistent information. More complex situation is shown in Figure 3. If one radio button is checked then the year options associated with other radio buttons are turned off to prevent choosing a wrong year.

- **Cross validation between different form questions** This checks the consistency of two or more answers to related questions. For instance, the relation between the age of a user and the year of arrival in Australia - a user cannot be younger than the period of their stay in Australia.
- **Field Type** Each form item is validated for the correct field type (alpha, alphanumeric, numeric etc.)
- **Field Structure** Fields such as email addresses or telephone numbers must exhibit a prescribed syntax.
- **Muti-stage Form Submission** This ensures that the users have properly stepped through the process of completing the form and they have digitally signed the form before their record is saved/updated in the system database.

4) Date of Birth: Day  Month  Year

5) In what country were you born?  
 Australia  Overseas

Your country name:

Year you first arrive:

6) Do you speak a language other than English?  
 No  Yes

Name of your language:

Country list (dropdown):  
Spain  
Switzerland  
Former Yugoslavia not further defined  
Croatia  
Slovenia  
Other European countries not listed above  
China, People's Republic  
Hong Kong  
Japan  
India  
Indonesia (including Timor)

Fig. 1 - Section of the Form capturing overseas country of origin for international students.

**8) Do you have a disability, impairment or long term medical conditions which may affect your studies?**

Yes  No

*Please indicate the area or areas of impairment:*

Hearing     Mobility     Medical

Learning     Vision     Other

*Would you like to receive advice on support services, equipment and facilities which may assist you?*

Yes     No

**Fig. 2 - A section of the Enrolment Form using JavaScript validation program (disability section)**

Users fill out the e-forms interactively by choosing different form items and responding to prompts. In the on-line enrolment system, the interaction is implemented by running JavaScript programs on client/user machines. This arrangement allows an efficient validation of data supplied by students. An alternative method of submitting partially completed forms to the server is time and resource consuming.

**This section (Q.14 to Q.19) asks you what education had you completed or commenced before you first enrolled in a course at UWS Macarthur.**

**14) Postgraduate course (higher doctorate, PhD, Masters, preliminary or qualifying, Post Graduate Diploma Post Graduate Certificate, etc)**

Completed all the requirements for the award in any such course in

Commenced, but not completed any such course in

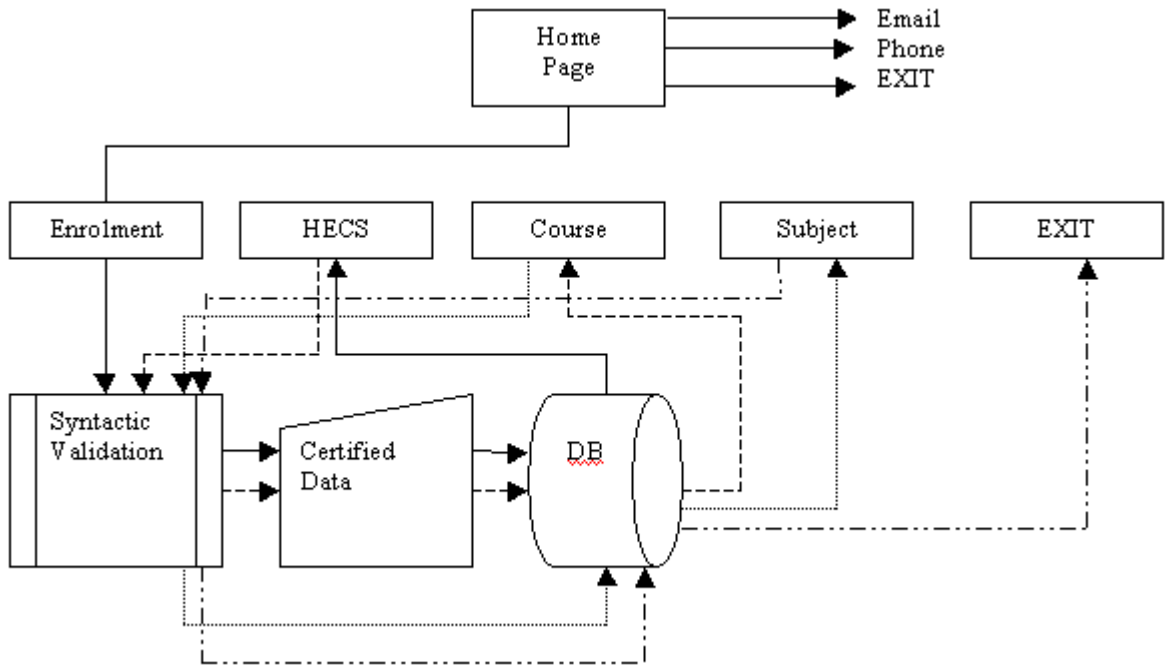
Never commenced any such course.

**Fig. 3 - Example from the main form showing the interactivity of form elements for validation.**

### 3. ON-LINE DATA ACQUISITION SYSTEM

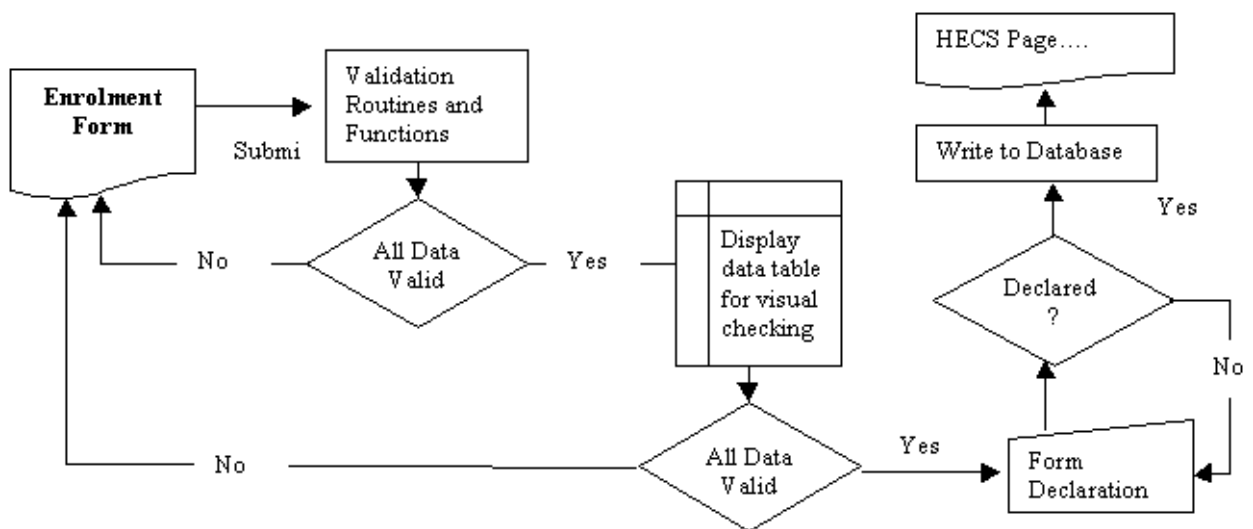
#### 3.1. General Structure

The prototype of our on-line enrolment system consists of many Web Pages, linked to data validation programs and aimed to collect specific student data. A general structure of the system is presented in Figure 4.



**Fig. 4 - Server Site Structure - initial stage**

Figure 4 shows an overall simplified structure of the system. The system allows students to navigate the site through its home page, which contains instructional information for completing the enrolment procedures, the students rights and their obligations and the contact details of the staff who can assist students by telephone or by email. This page has five main buttons: *Personal details*, *Course*, *Subjects*, *HECS* and *Faculty* information. Students would start their journey by initially completing their enrolment details. At this stage, this is the only menu button that would respond to the mouse click or pressing the "Enter key" on the keyboard. To record the entered enrolment details, students have to certify that the information entered by them into the form is true and accurate. Without their certification, the data will not be saved into the system database. Then students are guided to the next step. Figure 5 illustrates the collection of options available to users. This continues until all stages are completed. When this is done an electronic confirmation message would be send out to the student indicating that they now have completed enrolment and their data has been recorded in the system



databases.

**Fig. 5 - Form processing flow.**

This system can be further extended to provide other user services/applications such as Pre-enrolment Services, where students would be able to specify their intended Load (subjects) for their future year of study. Hence the continuing Load would be calculated. Based on this information universities would be able to better predict their annual student intake, plan places per course and allocate other resources to their academic and research activities.

### 3.2. Different Methods of On-line Data Validation

Because of the sensitivity of the information involved as well as its strategic importance to the planning and business activities of the universities, different methods of on-line data validation are suggested.

*Local Distributed Method* - It utilises the client machine by running mobile code (in our case, the JavaScript Validation Program), on the user computer (see Figure 6-a). We assume that the client machine is trusted in a sense that its Operating System is administered by the University IT section. The validation in this case is fast. Unfortunately, our experience shows that a perverse user may bypass the Operating System safeguards, fabricate a form and submit it to the server for processing. To overcome this problem, we suggest to verify the form originality (this will be discussed in Section 5).

*Central Distributed Method* - It can be of practical use if the browser is part of a trusted Operating System. Because, presently this is not the case, the central method uses the Web Server processing power for validating the information submitted to the server in addition to verifications done at the Browser side (see Figure 6-b). This method will fully satisfy the correctness of data. However, it exerts a heavy load on the Web Server especially when hundreds of users are simultaneously connecting to the site at the peak enrolment period, hence the Web Server will be a bottleneck in the Enrolment System. So this method may not be a satisfactory option.

*Dedicated Distributed Method* - one or more computers are dedicated to perform validation services on behalf of the server (see Figure 6-c). This is similar to the central method with the benefit of not overloading the server. This would provide satisfactory results but it is an expensive solution and also carries an additional maintenance and administration overhead. Therefore this may not be an economical solution when compared to the above two methods.

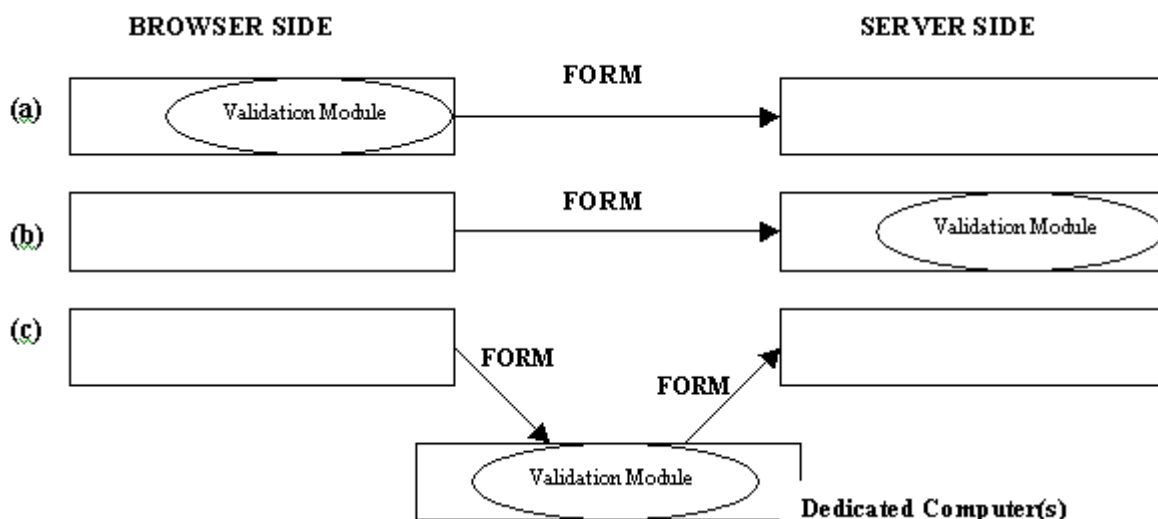


Fig. 6(a, b and c) - Different Data Validation Techniques.

### 3.3. Dissemination of Data

After students have successfully stepped through the data entry and validation processes, they will try to submit the form to the server. Prior to the final submission, a screen containing the table of entered data is displayed asking the students:

- to digitally sign the form and certify the correctness of the information; or
- to review it once more, correct the detected mistakes and then sign the page.

When the form has been signed, the data fields in the form are transferred into a database on the server. The Active Server Pages (ASP) program acts as a mediator between the data fields in the form and the server database. It converts the English content of data fields supplied by the students, except the contact details, into the alphanumeric codes specified by DETYA [2, 3] and records them in the student database located in the server. This is the basis for production of ASCII files, which are used as the input for:

- Student Record System to populate its data fields,
- the DETYA developed software, DETYAPAC, for detailed cross validation before submitting the files to the government;
- statistical packages like SAS, spreadsheets and different database for statistical analysis purposes in a readily available format;
- the same forms, which once filled out, can be viewed and corrected, if some data (like home address or other contact details) have changed.

### **3.4. Characteristics of the prototype On-line Enrolment System**

The system offers advantages including the following:

- Capturing accurate and mainly error free data;
- Reduced cost compared to paper based systems;
- Proper implementation guarantees a proper balance between security (privacy) requirements and availability of information (data).
- Availability of enrolment data in a format suitable for data analysis;
- Timely and accurate update of the students records in SRS;
- On-campus availability of the system to all students;
- Easy to navigate internally and externally;
- Creating a corporate presence on-line;
- Better client oriented services;

However, there are certain limitations in using such a system including the following:

- Internet access to some students outside the university. A recent university survey of commencing students has shown that 70% of students have access to Internet outside the university [19]. Although Internet access each day becomes more widespread, presently, this could be counted as a limitation, despite 100% Internet access within the University for all students.
- A probability of failure during accessing the system if the communication is of low reliability;
- Possibility of communication drop out while transferring data to the server;
- Costs associated with security;
- Changing format and medium of the required information for some organisations (e.g. ATO);

## **4. SECURITY AND PRIVACY ISSUES**

### **4.1 Classification of Data**

Normally the collected data can be classified into the following categories:

*Private or Personal Information:* All personal details collected on-line are treated as confidential and will not be revealed to unauthorised personnel. The University may use the information to prepare statistics required by the Federal and State Governments for funding and planning purposes. The University may be legally bound to report students' debts to the Australian Taxation Office (ATO), which keeps track of the Higher Education Contributions Scheme.

*University Owned:* The University could also utilise some information that enables it to better provide services to students or to conduct its marketing campaign. These could be classified as the University Owned data. However, the anonymity of the answers should be maintained and the results should be published in a manner that does not identify any individual.

*Statistical / Public Domain:* Some of the collected information could be compiled in a collective manner to provide public information on University's activities in the region. These could be information such as recruiting students to its different courses and other services, its community involvement, the composition of its students, and its attention to diversity and providing access and equity to its clients.

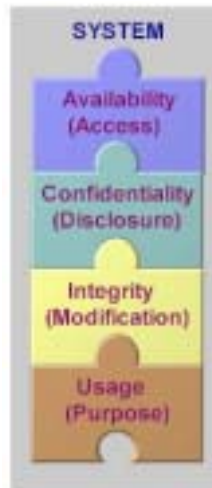
## **4.2 Privacy Issues**

The home page of the site contains instructional information for completing the enrolment procedures, student rights and their obligations as well as information on the usage of the collected information. Collecting personal details and other enrolment information on-line brings forth the issue of privacy and our obligation to users, as the provider of such an on-line system, to maintain the confidentiality of the users' information. This should be accomplished by not only implementing security standards in the web server, web browser and the communication between the server and the browser, also by preventing any unauthorised access to such information when available in electronic format. The users should also be informed how the collected information will be used and what organisations would utilise it. This would then require us to closely classify the information.

## **4.3 Security vs Risks and Possible Attacks**

*What is Web Security?* The concept of Web security has been perceived differently by users, organisations, Web Masters, Software vendors and the media [16]. Our students, as users of the system, require assurances that their communication sessions are private and the submitted information remains confidential. On the other hand, universities are demanding the ability to confidently conduct the highly sensitive enrolment processes on-line and that the submitted information is tamper free during the communication phase. Universities also require that their servers and site contents are intact and fully protected from unauthorised access.

Our security objectives are to find a situation in which factors such as *confidentiality* of the information, its *integrity*, its *availability* and the *purpose* of collecting the information is clearly defined, properly maintained and these are commonly applied in all aspects of the System's operation. Figure 7 shows the relationship of these components with the overall system.



**Fig. 7 – Relationship of the system with security components**

*What are Risks?* Because of the open structure of the Internet there are certain risks that affect both browsers and web servers. Risks are combinations of the following components:

- 1) *Threats to the systems:* Possible attacks to both systems, which could be attempted by :
  - Outsiders: hackers, attackers, impersonators, eavesdroppers, web spoofers;
  - Insiders: students, employees (normally authorised users exploiting bugs in system software or poorly configured privileges to set up backdoors for their subsequent intrusions);
  - Virus attacks;
  - Malfunctioning: Interruption of network services between the browsers and their hosts (user machines), between the browsers and web servers and also communication dropouts.
  - Natural Disasters
  
- 2) *Vulnerabilities:* These are mainly caused by system weaknesses such as inappropriate system configuration, network configurations, not up-to-date version of the operating systems and other software in use and not applying the hot fixes and service patches/releases provided by relevant software vendors.
  
- 3) *Assets:* Information and resources of critical importance. These should not be stored in the same partition as the public Web Server is located. It is preferable not to store them on the machine that hosts the web site.

To minimise risks in our On-line Enrolment System, many precautions and safeguards are undertaken. The configuration of the operating system has been tailored to support the main activity only. Network drives and printers have been removed from the web server. To overcome communication problems between browsers and the server, an independent communication buffer has been installed.

The interaction between a student who wishes to enroll and the Enrolment System (the server) progresses as follows. The student logs into the server via their browser. The browser fetches a copy of enrolment form and displays it on the student machine. The student enters the required data, asks the server for validation, signs the form and submits it to the server. Note that there are two parties: students and university whose security concerns are different.

Students assume that the server site is authentic (the server is expected to identify itself), the transmitted pages to their browsers are original (owned by the university) and they are free from any malicious mobile code. Students also believe that the server will not distribute the information, which are deemed to be confidential.

On the other hand, the university (the server) assumes that the students are acting in a good faith and want to enroll themselves. The server expects students to identify themselves before they are allowed to proceed with enrolment. Ideally, the server may also assume that students are friendly -- they do not intend to hack into the system, bypass the validation routines and cause the server to crash rendering the service unavailable to other users.

Both parties believe that the network provides the expected communication facility and is free from intruders who may wish to hijack the session (web spoofing) [5]. The information flow between browsers and the server is protected against any attempt of tampering, modification, and replay. Additionally, the parties may also want to have an option for the exchange of information via confidential channels. Now we discuss briefly the browser (client), the server, and the communication security.

#### 4.4 Browser Security

The two most popular browsers in Internet industry are Microsoft Internet Explorer® and Netscape Navigator® /Communicator®. In terms of security standards they are available in two versions. The US version of these products incorporates strong cryptography with long cryptographic keys (128 bit), while their export version for use outside US (International Version) applies short keys that are easily crackable by the brute force attack. This is also the case with the web servers sold around the world. Therefore if the security of either piece of software is compromised, the whole session is rendered insecure. The two mentioned above types of browsers offer different security configurations and features. Table 2 summarises the security features of the two types of browsers [11,17].

**Table 2: Comparison of security features in Netscape Communicator and Internet Explorer Browsers**

<i>Security Features</i>	<i>Microsoft Internet Explorer®</i>	<i>Netscape Communicator/Navigator®</i>
<i>Encryption and Authorisation Support</i>		
SSL	●	●
Basic Authentication	●	●
Certificate Authentication	●	●
Secure Electronic Transactions (SET)	●	
Cookies	●	●
<i>Active Contents</i>		
JavaScript (Enable/Disable)	●	●
Java (Enable/Disable)	●	●
ActiveX (Enable/Disable)	●	●
Active Content Permit/Deny per source computer	●	
Object Signing		●
<i>Other Security Features</i>		
Platform for Internet Content Selection (PICS)	●	
International 128-bit	●	●
Object Linking and Embedding (OLE)	●	
S/MIME	●	●
Central Control		●

Both browsers have similar security features. Normally user preferences are the governing factor for choosing either product unless certain system requirements specify otherwise. Netscape communicator allows users to sight the certificate of the active content publisher and also to edit the privileges of the applet and scripts signed by their developers. Netscape authenticode allows tracking down the malicious control and identify the author, while Microsoft suggests examining the document cache to identify the most recent controls run on the user machine.

The browser security is designed by carefully testing the Web Enrolment Software, to the extent that has been developed, for correct operation and viewing and as it has been intended for.

*Impact of Browser Settings on System Security* -Considering the restrictions that browsers will impose on execution of mobile code by allowing the users to disable this functionality, it will affect execution of JavaScript program to validate data. To overcome this problem, the system intelligently detects the browser for capability to run JavaScript code. If it has been turned off, it sends a warning message to the user to enable mobile code execution to safely continue with the intended on-line processing tasks. The browser, as a user front end, plays an important role in identifying the user to the host. Section 4.5 discusses this in details.

#### **4.5 User Identification**

Identification of users is usually one of the first safeguards, which is used to protect computer resources against an unauthorised access. There are many different types of identification techniques, from the low cost implementations (typically based on passwords) to very expensive and sophisticated mechanisms (normally based on dedicated identification devices). The Enrolment System uses passwords for user identification. The identification is performed twice on:

- *Operating System (OS) level (Server login and password) and on*
- *Application level (Enrolment System login and password).*

As soon as a user has completed the entry of the site URL (<http://ezenrol.macarthur.uws.edu.au/enrolME>) in the address bar of a browser, a login dialog box appears on the screen, asking for username and password.

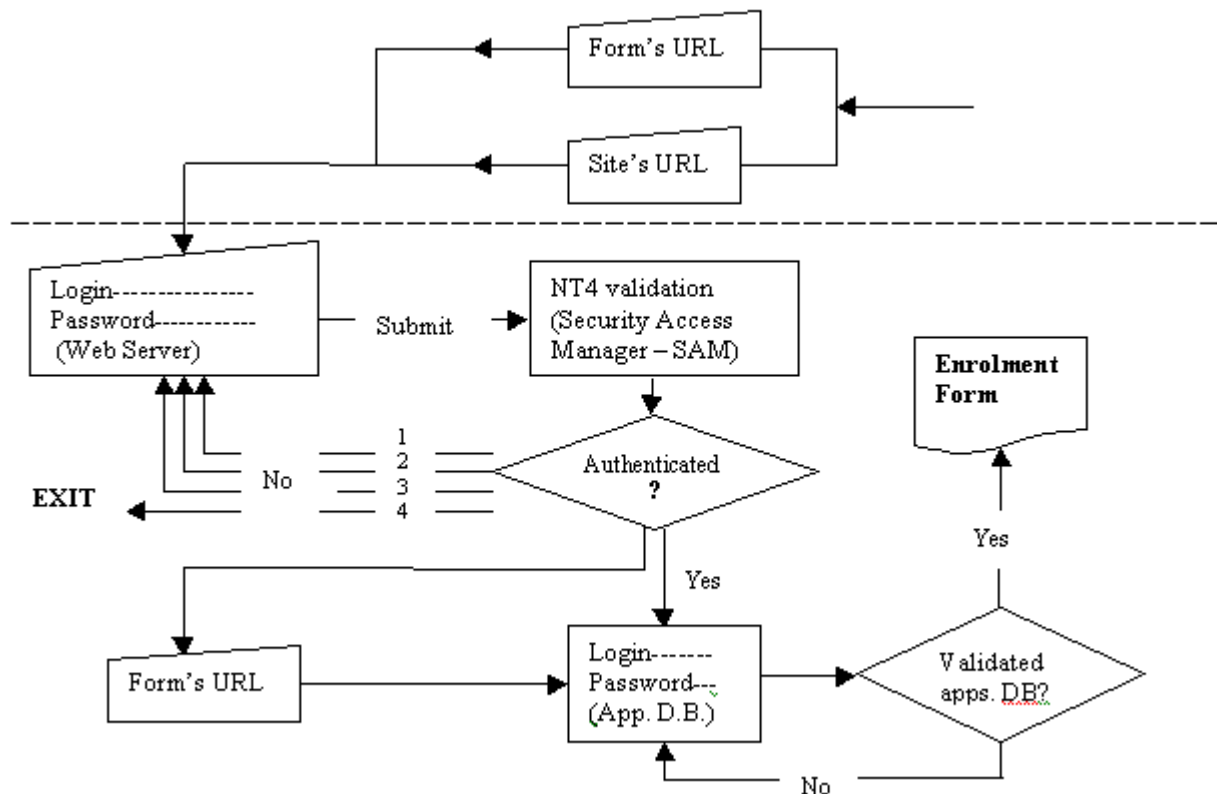
This is the mandatory web server account identification level. The entered details are then compared with the Security ID (SID), Group ID and User Rights Permissions stored in the system Security Access Manager (SAM) database, where each user has a pre specific information for identification purposes. After a successful user identification, the system NT4, generates an access token and switches into the user mode to permit the user to access resources through the Access Control List (ACL) which has been specified for that user in the system [13].

Next, the URL is opened and the user enters the second level of the identification. This is done by programs, which access the Web Application Authentication Database. This database contains username and password and security seeds for each user. The database authentication system is very efficient and easily allows addition and deletion of users details [1,9,21].

Hence the users should pass the two steps successfully before they obtain any access to confidential information forms. When successfully logged in, users can navigate the site and follow the instruction to complete their enrolments on line or making changes to the previously displayed details in the form to update their record.

The password used in the Enrolment System is a concatenation of the student date of birth (6-digit format - day, month and year 2 digit each) followed by a sequence of 6 to 8 characters selected by and known to the student.

The Enrolment System uses a mandatory startup page. This is an important security measure to get the users always to start site navigation from the login page. This prevents entering the site by simply typing the pre-recorded addresses (URLs) of the site pages/forms in the browser address bar, in order to bypass the identification page. This requirement remains in force even if the web server has successfully identified a user at the OS level. The Figure 8 illustrates the user identification.



**Fig. 8 - Login Procedures and associated procedures**

#### 4.6 Server Security

The Web Server is a complex system comprising many different components, each of which has its own security issues. The operating system binds these components and is itself a critical component of any web server. Therefore the choice of the operating system is important, which is typically restricted by the already used operating systems. UNIX and Windows NT are the most common web server platforms, which contain the required functionality. However, some application programs are platform dependent, although software developers try to write their programs to suit both operating systems. The UNIX power and its complexity, is a double edged sword - it is more flexible but also its security is difficult to verify especially in the context of the web security [23]. On the other hand, less complex systems like MS Windows or Mac OS may offer better web security. For instance, the Macintosh OS does not have a command interpreter and does not run any network services [16] so a potential attacker is restricted in their illegal activity. Therefore there is a tradeoff between convenience and security. Experts in computer security believe that the DOS operating system has set the security state-of-the-art back 25 years, and Microsoft has continued that legacy to this day [14]. But the real cause is more subtle.

The bottom line is that there is little to distinguish the two major operating systems, UNIX and NT, based solely on Web server application support. This choice is much more dependent on the environment in which the Web server is to run, the back-end data to be served, the technical support and programming staff, the client environment and the existing infrastructure. Some of these may even become irrelevant,

based on the industry trend to better integrating these two operating systems with products such as OpenNT [23]. Although for the prototype Enrolment System, we have used the Microsoft Windows NT 4.0 operating system, it may be preferable to use the UNIX operating system for easier connection to Student Record System databases, which are UNIX driven. This would also provide easier maintenance and support for the web server. On the other hand, one may also argue that from security point of view, it may be better to run web servers on totally different and isolated operating systems than those on which the corporate applications are running.

The Enrolment System includes a trusted and untrusted area. The OS, databases, security information and critical applications of the system are placed in a different partition, under high security policies and security configuration (trusted area), while the actual web server software is running in untrusted area.

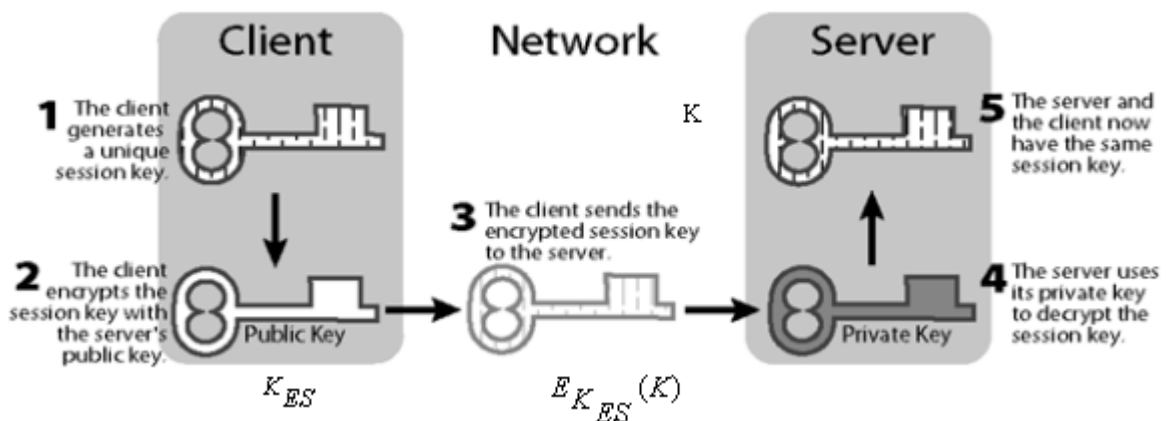
To secure the system we use two-level user identification and a mandatory start page and the web server is configured to protect it from break-ins, site vandalism and denial-of-service attack. In the prototype, the web server is run behind the University firewall. Measures have been taken to monitor security and identify intrusions such as sniffing attacks. We have also continue to update the operating system and server software and keep installing hotfixes and other relevant OS patches as a part of our security policy.

Also we have installed virus detection software with the capability of periodically updating its Virus Dictionary with the latest information to combat viruses. This assures that our site would not pass any virus to the users system. However, students should also take steps to equip their systems with the virus scanners and also configure their Browsers with the recommended standards. For on-campus browsers, they have been configured so that the integrity of the Local Network can not be compromised.

The On-line Enrolment server operates in the university's network; hence certain security policies associated with the network environment should be established. The following section briefly describes this issue.

#### 4.7 Communication Security

Protection of data flow between browsers and the server is done by using cryptographic operations offered by Secure Socket Layer (SSL). The protection (authentication and/or confidentiality) can be based on either public-key or private-key cryptography [18]. Typically, the communicating parties use each other certified (authentic) public keys to run the Diffie-Hellman key agreement protocol. If successful, the protocol allows the parties to share the same (secret) key, which can be later used to establish authentication and/or confidential channel [15]. Figure 9 shows the key exchange between the server and the browser.



**Fig. 9 – Key exchange in a Secure Web Environment** (source: Stronghold (1997) [18])

Network Configuration – presently access to the Web Server has been confined to a specific data port and certain network ring on the university's network for further security control of the system.

This setup is suitable for on-campus students' access to the system on university's Virtual Private Network (VPN). However, to make the system globally available to every student, we should implement other solutions. An example would be running the VPN behind a Firewall and utilisation of Proxy Web Servers for access to other Internet services as a separate activity away from the VPN.

The Internet uses the TCP/IP protocol for data communication. This protocol was not designed to provide any security; however, now it becomes more evident that communication security is a must in the Internet.

To rectify this omission, the Internet Engineering Task Force (IETF) has proposed a new version of TCP/IP called IPv6. Among its new features, it provides confidentiality and data integrity checking at the network layer of the ISO reference model, a facility known as IPSec (IP Security).

The protocol supports two modes of encryption: transport and tunnel. In the transport mode, only the data portion of each packet is encrypted; while in the tunnel mode, both the data and header information are encapsulated within the encrypted data stream (this is normally the feature of Virtual Private Networks).

Unlike SSL, IPv6 does not require the software to be modified in order to take the advantage of its encryption. It is expected to be faster than SLL as it was designed to be supported by the coming generation of IPv6-compatible hardware routers. However, because it runs at the network layer, it can not support user authentication, which normally occurs at the application layer, while it authenticates machines, routers, bridges and other network interfaces.

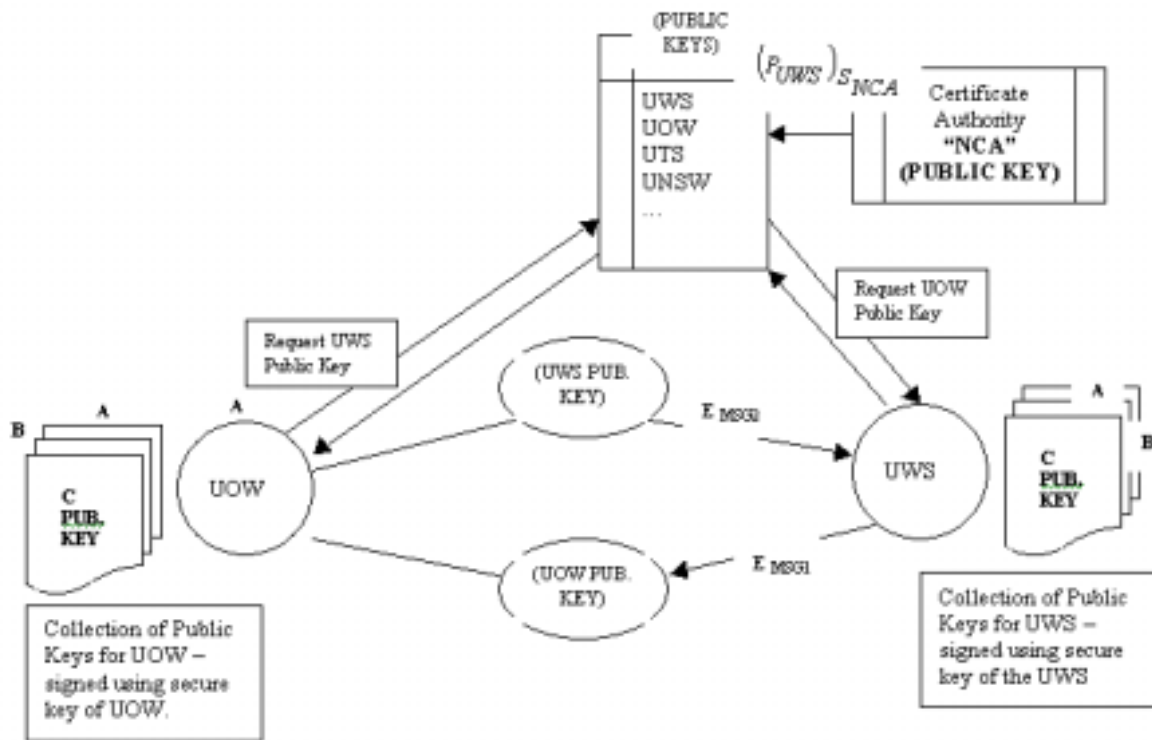
Another problem with this protocol is its dependency on certain hardware, therefore it will not become widespread until the time that right hardware is available for it in the market. For these reasons, there will continue to be a role for higher-level protocols, such as SSL and SET, in the IPv6 world [16].

Another crucial (and not solved yet) aspect of communication security is the inter university public key infrastructure.

To establish a secure communication channel, the two parties involved in a conversation have to know each other's authentic public key. There must be a trusted authority whose public key is known to everybody and it is responsible for maintenance and distribution of public keys in the form of certificates (a public key signed by the authority).

This authority may be either a Certificate Authority (CA) formed within a university system or a National Certificate Authority (NCA) for all universities within Australia. Either way, a public key of the authority must be published so everybody will be able to verify certificates of public keys. This authority displays public keys for its clients (note that clients retain their full control over their keys).

A proposal for an inter university public key infrastructure is illustrated in Figure 10.



**Fig. 10 - NCA and Public Key Infrastructure in a higher education setting.**

The NCA publishes its public key  $P_{NCA}$  normally in newspapers. This key is valid for 6 months to one-year duration. The matching secret key,  $S_{NCA}$ , of the NCA is used to sign all other public keys e.g. University of Western Sydney,  $(P_{UWS})_{S_{NCA}}$ . A user can get the public keys from NCA by copying the suitable entry and verify whether the entry is valid (this is done using the public key of the NCA ( $P_{NCA}$ )).

Formation of a National Certificate Authority for Universities in Australia is the most cost effective and easy to administer option than obtaining Digital Certificates from Companies like VeriSign [22] or Thawte [20] in USA, which bear significant costs. This infrastructure not only enables secure communication and transfer of confidential information between students and universities, it also enables collaborative work and exchange of research and other important documents between universities in a secure and more reliable environment.

## 5. VERIFICATION OF FORM ORIGINALITY

The problems occur when a perverse user tries to send information to the server by fabricating a forged form containing some or all of the data fields as in the original, bypassing all validation routines and then submitting the form using the same action URL as in the original form. This is possible due to the stateless nature of HTML, where each resource requested by a browser is independent of any previous history, and the openness of HTML and JavaScript, the source "code" being readily readable by the browser. Although such behavior depends on the skills of an illegal user, with a large community of users, it can be anticipated that such security violations may occur. There are *two distinct problems* related to the form verification: verification of data syntax in the form (we are going to call this type of verification, form validation) and verification of form originality (authorship).

## 5.1. Validation Methods

Validation of form originality can be conducted according to the following methods:

1. *Duplicating the validation of all the data fields in the form at the server side* - Although this is conceptually the simplest solution, it may not be an efficient solution, producing additional server loads at time of peak server usage, namely the enrolment period of thousands of students.
2. *Selective validation of important fields and/or sections* - This is really a smaller version of the above, which still shares the same problem but to a lesser degree.
3. *Validation based on a good knowledge of the user* - In this situation, and in any similar one, the user has already had to pass some form of registration/authentication by either password/user ID, restricted or known IPs, or similar [1]. This means that the system can validate a form by checking the data characteristic (and hopefully confidential) to the user. The underlying assumption is that a user, knowing that they can be identified, will be well behaved. Establishing the security of the communication link to the known user is relatively easy [1,9,21]. The problems of relying on this method lie in:
  - a) Making sure that the user is aware that they are known;
  - b) Further validation is still required since corrupt data may still be entered with the associated side effects to the system;
  - c) A user can simply deny their perversity with the statement they followed directions and it must have been the system;
  - d) The user may not mind being identified! Legally or otherwise, it would be highly unlikely that any action could be brought against them. In any case they may not even care.
4. *Use of a "signature" that can be submitted only from the original form (using combination of Cookies, storing a small amount of data on student's computer, and JavaScript techniques)*

The concept, in this case, is that if a unique digital signature can be associated with the original form and it cannot be constructed easily by a duplicate form on another server, then this can be used to reject any "perverse" form and its data. The problem here is that the openness and stateless nature of HTML does not lend itself to a secure method. By using a combination of cookies sent in the original form header that can then be used by a JavaScript algorithm to generate either another cookie in the action URL header or by an additional form element a signature can be generated. However, the JavaScript will still indicate on reading that cookies are being used and the workings of the algorithm. A perverse and knowledgeable user can write their own scripts to extract cookies from the original form header and together with the knowledge of the algorithm produce the required signature. This does, of course, filter out the less knowledgeable users. On the other hand cookies may not be able to perform the task, if the user browser does not allow Cookies activity on their computer (as discussed in Client Side Security section of this paper).

5. *Presenting the whole Form as a Java applet with an inbuilt algorithm to produce a unique signature*

This is an extension of the above idea. Since it would require considerable effort for a perverse user to disassemble the Java byte code of an applet to determine the signature algorithm, this method has much promise.

6. *Examining the user's History Object for the address of the original Form sent to the user.* *This can be easily fabricated by a perverse user.* The knowledge of such a method being again readable from the original form source. Secondly it maybe that the "UniversalBrowseRead" [6] of the user machine does not allow access to the History Object.

### 7. Use of an ActiveX control downloaded to the Client to generate a signature

Although this method would certainly give a fairly protected signature, it assumes that the client is using Microsoft Explorer or Netscape Communicator, ActiveX control has not been turned off and the environment is a PC type machine.

## 5.2 Form Validation Protocol (FVP)

An enrolment server (*ES*) displays a suitable collection of electronic forms accessible via its web site pages. The forms typically allow students to enroll, to amend incorrectly recorded enrolment details such as their personal details, submit requests for adding or subtracting subjects, requesting leave of absence, requesting advanced credit etc. A student can access forms using a browser, which on their behalf, copies a form requested from the *ES* so the student can fill it out. When the form is ready, the student notifies the browser that the editing process has ended and the form is to be submitted to the *ES*.

### 1. Assumptions and necessary Infrastructure

We assume that both the *ES* and the browser or more precisely, the client host, which is running it, is registered and all users can get their authentic keys from the trusted authority (NCA). Normally, the public keys are accessible in the form of publicly verifiable certificates. The verification is done using the unique and authentic public key of the NCA.

Each student *P* has an account in the *ES* host so the host stores the student password  $passwd_p$  in the form of its digest, i.e. the password file contains an entry ( $P, H(passwd_p)$ ) where *P* is the student's login name,  $H(passwd_p)$  is the digest obtained using *cryptographically strong hash function*  $H()$  with public description. The file must be accessible in the superuser (administrator) mode only.

### 2. Validation Protocol

The protocol is aimed to be no less secure than security based on password login. Some of the steps seem to be an "overkill" and we are looking at different ways to economise the design. This is a starting proposal and it is likely to change depending on an acceptable tradeoff between the security level and the computation overhead required by the protocol. The protocol proceeds according to the following steps:

- Student *P* logs into a client host *C*. It can be a student's own PC or perhaps a machine accessible on campus. In either case, the host must have Internet connection to the *ES* and must have its public Key  $K_C$  registered with the University's Trusted Authority (UTA) (or alternatively known to the *ES*). The secret key  $K_C$  is stored within the host. Next the student opens a copy of the browser available on the host.
- The browser (*B*) fetches a copy of the Form *D* from the *ES*.
- The student edits the form *D*. After completion, the browser takes the student password from the session variable in which it is stored at the time of login.
- Browser selects a sufficiently long random number  $r$ , appends it to the student ID, *P*, and the student password and encrypts the triplet

$$E_{K_{ES}}(P, password_p, r)$$

- Using the public key of the enrolment server *ES*. Next, it appends the cryptogram to the document *D* and hashes the pair, i.e. it gets the digest

$$h = H(D, E_{K_{ES}}(P, password_p, r))$$

the host signs the digest

$$sgc = S(H(D, E_{K_{ES}}(P, password_p, r)))K_C = S(h)$$

where  $S()K_C$  is the signing function.

- $B \rightarrow ES : D, sgc, E_{K_{ES}}(P, password_p, r)$  Or the browser sends the signed document along with the cryptogram  $E_{K_{ES}}(P, password_p, r)$  to the ES.
- The server ES:
  - First decrypts the cryptogram  $E_{K_{ES}}(P, password_p, r)$  and recovers the triplet  $(P, password_p, r)$ ,
  - hashes the password and checks whether there is an entry  $(P, H(password_p))$  in the password file. If the check fails, the ES aborts, otherwise continues,
  - verifies the signature on the document D, i.e. recreates  $h = H(D, E_{K_{ES}}(P, password_p, r))$  and compares it with the digest  $h'$  recovered from the signature. If  $h \xrightarrow{?} h'$ , the document is considered genuine, otherwise invalid.

Two situations could occur:

1. If the Form originality correctly passed, students would be able to save their particulars in the site's database;
2. If it fails, the server sends a message to the student explaining the occurrence of the error.

Normally if students are correctly authenticated for gaining access to the system, no problem is expected to arise unless after entering the site they try to bypass any mandatory requirements.

## 6. CONCLUSION

Web computing, as an attractive, efficient and cost effective technique, can be utilised for accurate data capturing and on-line validation applications in a university setting. Confidential information can be transferred to the storage media via a secure communication channel. However, the openness of both HTML and client side scripts introduces an interesting problem of maintaining data integrity in the case of perverse clients. Sufficient measures can be put in place to ensure that the original form and its contained data fields have been submitted by the right people and are fully protected from those who may tamper with data or try to fabricate the forms. An alternative method to the use of *Form Validation Protocol* is duplication of validation on the server side, which would create server bottlenecks at the peak enrolment time.

This development, on the other hand, may bring up disconcert with other organisations' systems, which have not yet employed this technique and/or their preferred method of dealing with their clients who are also our clients as well. However, this can be worked through to obtain satisfactory results.

The implemented security techniques, system's user-oriented and user-friendly environment, validation of the data on the spot, elimination of later data entry procedures, and more importantly system's availability to users for updating their particulars as these may change would win the confidence of the students to enter their confidential information on-line. However, the Internet may not be reachable by every student outside the universities. The survey of *commencing students* at UWS has shown that about 30% of our students do not have access to Internet outside the University. Therefore the system, currently, can not cover everyone.

Web applications bear the risk of being broken into and security as such does not guarantee its safety. However, implementing all necessary security measures make it difficult for an attacker to find easy access to the information. Attackers would be further discouraged by higher criminal penalties.

This work is continuing research and as yet is not conclusive. We intend to present the latest happenings and results of our findings in other forums and conferences to the Higher Education community. Government organisations like DETYA may wish to consider support for complete development of the On-line data acquisition system as an appropriate vehicle for inputting data to its data compilation software, DETYAPAC. The data collected via this system can significantly improve the confidence in collected data for statistical and planning purposes.

## REFERENCES

1. Berry, D. (1998). *The Dynamic Authentication Filter* [Internet]. 15Second, Internet.com Corp. Available from: <<http://www.15seconds.com/Issue/980620.htm>> [Accessed June 20, 1998].
2. Department of Education, Training and Youth Affairs – DETYA. (1999). *Technical Requirements for Student Data Collection*. Canberra, Australia: Commonwealth Publishing, DETYA.
3. Department of Education, Training and Youth Affairs – DETYA. (1999). *Student Data Collection Manual*. Canberra, Australia: Commonwealth Publishing, DETYA.
4. Department of Education, Training and Youth Affairs – DETYA. (April 1999). *Quality Assurance and Improvement in Australian Higher Education – a summary*. Canberra, Australia: Commonwealth Publishing, DETYA.
5. Felton, E.W., Balfanz, D., Dean, D. and Wallach, D.S. (1997). *Web Spoofing: An Internet Con Game*. In: Proceedings of the 20<sup>th</sup> National Information System Security Conference [Internet] October 1997, Baltimore, Maryland. pp. 95-104. Available from: <<http://csrc.nist.gov/nissc/1997/proceedings/>> [Accessed Jan 1998].
6. Flanagan, D. (1998). *JavaScript, The Definitive Guide*. 3<sup>rd</sup> ed., CA, USA: O'Reilly and Associates, Inc.
7. Gollmann, D. (1999). *Computer Security*. England: John Wiley & Sons Ltd.
8. Hanneghan, M. (1996). *The World Wide Web As A Platform For Supporting Interactive Concurrent Engineering*. In: Proceedings of the 8<sup>th</sup> International Conference of Advanced Information Systems Engineering, Springer, LNCS 1080
9. Mnemonic, J. (1997). *Access Control With A Databas* [Internet]. Available from: <<http://www.asphole.com/asphole/default.asp>> [Accessed Oct. 26, 1997].
10. Murugesan, S., Deshpande, Y., Hansen, S. and Ginige, A. (1999). *Web Engineering: A New Discipline for Development of Web-based Systems*. In: *Proceedings of the first ICSE Workshop on Web Engineering, International Conference on Software Engineering, 16 -17 May 1999, Los Angeles, USA*. pp. 1-9.
11. Netscape Netcenter. (1999). *Netscape Security Features Evaluation Guide* [Internet]. Netscape. Available from: < <http://messenger.netscape.com/products/security/resources/evalguide/>> [Accessed March 25, 1999].
12. Rubin, A., Geer, D. and Ranum, M., 1997, “*Web Security Sourcebook*”, John Wiley and Sons.
13. Russel, C. and Crawford, S. (1997). *Running Microsoft Windows NT Server 40*. USA, Microsoft Press.
14. Schneier, B. (1999). *The Trojan Horse Race*. Communications of the ACM, 42 (9) September 1999, p.128.
15. Stein, L. (1999). *The World Wide Web Security FAQ* [Internet]. L. D. Stein. Available from: <<http://www.w3.org/security/Faq/>> [Accessed Sept. 13, 1999].
16. Stein, L. D. (1998). *Web Security*. 2<sup>nd</sup> Print, Massachusetts, USA: Addison Wesley Longman, Inc.

17. Stewart, J. N. (1998). Microsoft and Netscape – A Focus on Security . *WebServer Online Magazine [Internet]*, May 1998 issue. Computer Publishing Group Inc. Available from: <<http://webserver.cpg.com/ws/3.5/>> [Accessed Jan. 19, 1999].
18. Stronghold (1997). Keys and Certificate [Internet]. C2 Net International. Available from: <<http://www.int.c2.net/support/spwp/docs2.0/cryptography.html>> [Accessed Feb. 28, 1999].
19. UWS Nepean Development and Information Management Planning Services, DIMPS. (1999). Student Intake Survey 1999: A Survey of Commencing Students at Semester 1 and 2 Enrolment. Sydney, DIMPS unit.
20. Thawte (1997). SSL Certificates [Internet]. Thawte Consulting. Available from: <<http://www.thawte.com>> [Accessed Feb. 18, 1999].
21. Trotter, A. (1998). ASP Authentication Using IP Address [Internet]. 15Seconds, Internet.com Corp. Available from: <<http://www.15seconds.com/Issue/981104.htm>> [Accessed Nov. 04, 1998].
22. VeriSign (1998). SSL Server Certificates [Internet]. VeriSign Inc. Available from: <<http://www.verisign.com>> [Accessed Dec. 19, 1998].
23. Westmacott, I. (1998). The UNIX vs. NT Myth. SunExpert Magazine, June 1998, pp. 68-71.

## ACKNOWLEDGEMENT

The authors wish to thank Dr San Murugesan and Mr. Peter Manass for reading the manuscript and their helpful comments. Thanks to Mr. Damian Hawkins and Mr. David Cawthorne for offering form design ideas by always keeping the users of the system, students, in mind. We would like to thank Ms. Anne Barrie for providing us with the draft copy of the Student Intake Survey 1999. Also thanks to the UWS Macarthur Registrar, Ms Rennie Jackson for her foresight, encouragement and support of the project. Thanks to the Human Resources Division of the University for providing funding, through its staff development grant, to purchase the initial hardware and software necessary to set up the system.