VECIMS 2003 – International Symposium on
Virtual Environments, Human-Computer Interfaces, and Measurement Systems
Lugano, Switzerland, 27-29 July 2003

Design of an Access Control Module for an Instrumentation Gateway

Paul G. Greeff, Gerhard P. Hancke, Derrick L. van der Merwe and László Horváth
Department of Electrical, Electronic and Computer Engineering,
University of Pretoria, Pretoria, 0002, South-Africa.
Phone: +27 12 420 4334, fax: +27 12 362 5000, e-mail: paul.greeff@eng.up.ac.za

Abstract *- This paper describes some of the preliminary work performed on a* **Resource Access Control** *module for an instrumentation gateway. The access control module employs a variation of the popular RBAC (Role based access control) scheme described by various authors, including Ferraiolo [4][5][6]. The gateway has very little security, hence, any user able to log onto the system is able to control any resource. This module aims to implement a plugable module whereby users are granted access to various parts of the system depending on their user rights whilst giving the administrator a very powerful method of restricting access without sacrificing ease of administration.*

I. INTRODUCTION

The Computer Network and Security Group within the Department of EE&C, in collaboration with ICT TUV is investigating the concept of a smart card automation network. The idea is to create a decentralized control system with networks of intelligent nodes (refer to Figure 1). The idea forms the basis of the IGUANA (Internet gateway for unified automation network access – URL: http://www.ict.tuwien.ac.at/iguana/) project. The gateway provides access to a Field area network. Typically a user would connect to the ESD (Extended services daemon) through the Internet. The ESD software module queries the connected FANDs (Field are network daemon) to obtain available resources and/or services.

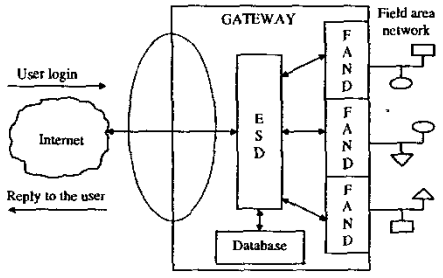

Fig. 1. The figure shows the current structure of the IGUANA gateway. Note that the dashed circle indicates the placement of the proposed access control module.

The problem addressed is the access control capabilities of the IGUANA gateway. Currently the IGUANA gateway possesses no capability of allocating resources and/or services. Any client may have access to any other client's resources or services.

The access control aims to successfully integrate with surrounding IGUANA software modules, while allocating the available resources and/or services of the ESD through the application of an access control scheme.

II. BACKGROUND

Various access control schemes exist, but vary in applicability depending on the application. Below follows a brief summary of some of the popular schemes.

A. Access Control List (ACL)

ACLs [4] are one of the most commonly used access control schemes. The basic principle behind ACLs is that every piece of data, application or service has a list of users associated with it. The association specifies that a user may, or may not, have access to the specific object. The system will prevent anyone not on the access control list from accessing the record.

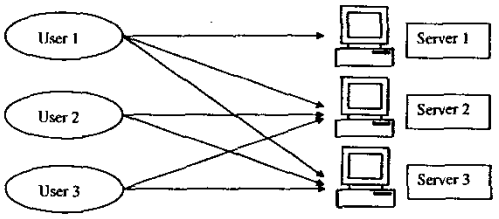

Fig. 2. The figure shows a model of access control list [3]. User 1 has access to all three servers, and the other users have access to server 2 and server 3.

Figure 2 illustrates the associations between users and respective objects. Each server may function as a data storage facility, application server or authentication server.

User 1 can be seen as a system administrator. The administrator needs to maintain the access rights of all the users. In this scheme, it is easy for an administrator to see which users have access to which data and/or applications. Each data and application server has its own ACL. The administrator simply needs to add or remove a user from the respective ACL.

The problem however multiplies whenever a person's role in the organisation changes and the amount of resources and services expands.

There exists a more efficient implementation, namely ACL groups [1]. ACL_G is an instance of ACLs where only groups are permitted as entries into an access list. Users are associated with a group and the group may be associated with specific permissions. The administrator's job is thus simplified.

B. Role Based Access Control (RBAC)

RBAC is an access control scheme that grants users access to information based on their responsibilities within an organisation [5]. Access permissions are associated with roles and the users are assigned to these roles, as shown in Figure 3. This scheme creates an opportunity for the administrator to express the access control policy as a logical implementation of how the organisation is viewed.

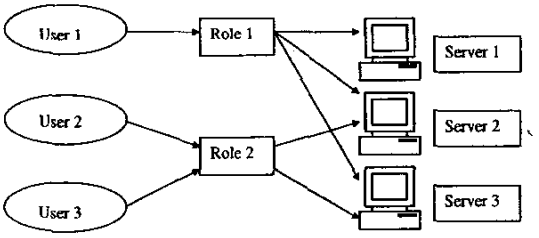

Fig. 3. The figure shows a model of role based access control associations[3]. User 1 is indirectly associated with permissions regarding all three servers and the other users are indirectly associated with permissions regarding server 1 and server 2.

There does not seem to be a great difference between Figure 2 and Figure 3, however imagine a network consisting of thousands of users and multiple servers. The task of managing an ACL scheme would be daunting.

The most important benefit of RBAC is that it simplifies the management of the access control policy. The administrator can control access to the system at a level of abstraction that is similar to the structure of the organisation. This feature creates a visualization of where, how and why certain complex access control policies are implemented. If the responsibilities of a user change, so do their role in the organisation. If user 3 were hired as an administrator, it would be a simple task of moving his account form role 2 to role 1. No metadata is kept for the decision-making process.

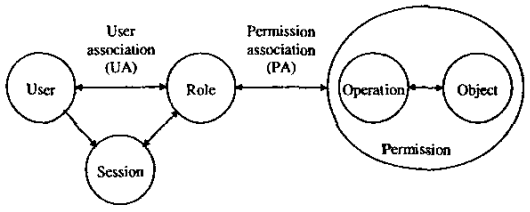

Fig. 4. The figure shows the relationship between the core RBAC model elements [6]. The double-sided arrows illustrate a many-to-many relationship. The single-sided arrows illustrate a one-to-many relationship.

To provide further administrative efficiency, RBAC allows roles to inherit permissions from other roles. This inheritance creates a role hierarchy [2]. For example, the role "physician" is the hierarchical superior of the role "nurse". The physician contains all the permissions granted to the nurse, including any new permission granted to the role "physician".

RBAC provides the capability to implement any access control scheme, through the use of heterogeneous applications, on a variety of platforms.

As can be seen in Figure 4, the Core RBAC consists of the following model elements.

User: A set of clients, both trusted and not trusted, who use the system.

Roles: A set of named duties or job functions within an organisation.

Sessions: A mapping between a user and an activated subset of authorised roles.

Operations: A set of access modes or actions permitted on an object.

Objects: A passive entity within the system, which must be protected against an attacker.

Permissions: A logical combination of operations and objects.

User association: An association between an element in the user entity and the role entity.

Permission association: An association between an element in the role entity and the permission entity.

A user may be assigned to many roles, but the session element ensures separation of duty. The session element specifies which of the authorised roles a user can occupy at once. The aim is to countermeasure fraud and other conflicts of interest.

Permissions consist of a set of operations that can be performed on a set of objects. Thus a role must be authorized to read/write from an object and the user must be assigned to that role. A role may be associated with multiple permissions.

The principle of least privilege also applies to RBAC. The principle states that the user only be given the necessary permissions to fulfil a specified duty; no additional permissions should be given.

The RBAC scheme has become the standard in access control schemes.

C. Other Access Control Schemes

Many other schemes exists, however the focus of this document is however on RBAC and ACLs

III. CONTRIBUTION

RBAC is an access control scheme with multiple advantages and very few limitations. RBAC does however possess two major weaknesses, namely in the fields of auditing capabilities and dynamic permission validation. Currently, as mentioned, the user is associated with many sessions and a session may be associated with only one user. The session entity however does not log any role activations and operations completed. The session entity [6] implements only the following session functions.

CreateSession: Creates the session and provide the user with a default set of active roles.
AddActiveRole: Activates a role in the current session.
DropActiveRole: Deletes a role from the current session.
CheckAccess: Check whether the user possess the requested permissions through role activation.

With the access control module the session entity consists of sub entities responsible for logging role activations and operations performed. A single query can conveniently retrieve information concerning any session associated with a user and/or role. Therefore you can detect an illegal entity accessing secure context sensitive data.

The second important issue addressed by the access control module is that of dynamic permission validation. Strangely enough Ferraiolo, Sandhu, Gavrila, Kuhn and Chandramouli [6] do not define any function associated with operation- and object creation. Similarly permissions are only discussed in permission associations with roles.

The access control module implements a completely new way of validating permissions, together with the standard create and delete permission scenarios. A query can be issued to the ESD that replies with a list of nodes and data points currently active on all the existing FAN daemons. The administrator may specify all the available operations offered by the ESD. After the operation and object entity has been updated the validity of all permissions can be checked and activated or deactivated accordingly.

IV. IMPLEMENTATION

A short summary of all the functional units in Figure 5 is given below.

FU1 is a collection of software functions that initialises elements of the access control module. These elements includes interface and thread variables. A collection of software functions in FU2 receives a user identification number from FU6 and creates an administrator or client thread. A software module, FU3, is used by a remote gateway administrator to configure, maintain and review the hierarchical RBAC policy. FU4 shows a software module used by a remote gateway user to allocate, on request, required resources. The gateway user chooses which roles to activate and operations to perform. FU5 is then used by the local gateway administrator to review system and MySQL database information. Data is then exchanged over a TCP/IP connection with the TLS (Transport Layer Security) [7] module using FU6 as an interface. FU7 issues queries to a MySQL server, receives the result and processes the result. FU8 exchanges data over a TCP/IP connection with the ESD module. FU9 inserts error information (interface, calling module, time and user) into the MySQL database. The interface unit mostly generates the errors. If an error is fatal FU9 calls FU10 to log out the user. FU10 is a collection of software functions that stops a specified user's thread.

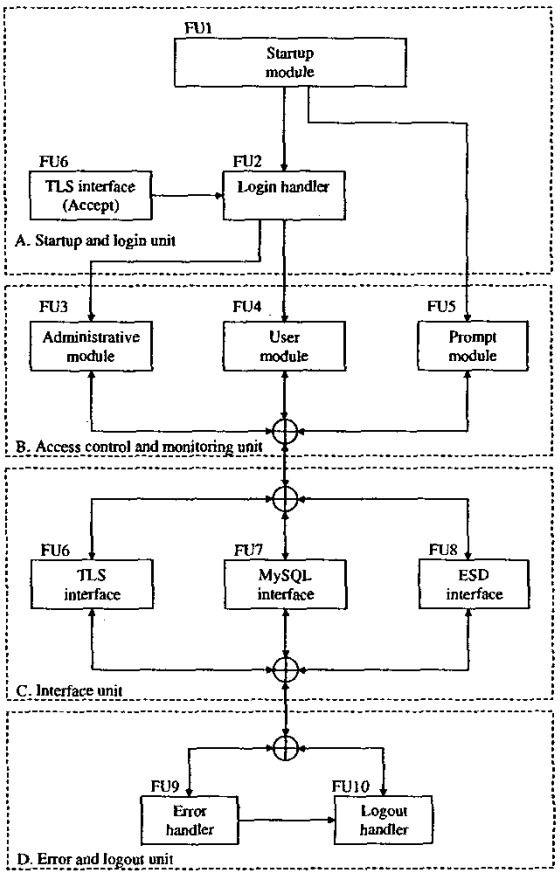

Fig 5. Functional block diagram of the access control module.

The access control module software consist of a full core RBAC command set with role hierarchies and DSD (Dynamic Separation of Duty), an ANSI-C administrative access control sub-module to ease the tasks of the IGUANA gateway administrator in configuring the RBAC scheme, an ANSI-C user access control sub-module to be used by the IGUANA gateway client in activating authorised roles and performing authorised operations, three ANSI-C interfaces to interact with the existing surrounding IGUANA software modules and the newly created TLS software module, and a test MySQL database (implementing RBAC) that can be used for simulation, training and development purposes.

A. Concept design

The concept design (Figure 6) resulted in the implementation of a RBAC scheme through the use of MySQL as a design tool. RBAC does require more processing than ACL, but through the intelligent use of MySQL table referencing and optimisation techniques, this problem could easily be overcome. The simple database management structure of MySQL also enhanced the logical organisational representation offered by RBAC.

MySQL makes it easy to manipulate different types of data. The security offered by MySQL is also a huge advantage - easing the task of protecting the database.

The advantages that RBAC possesses over ACL are that an organisation's hierarchical structure can be easily represented, it places a high emphasis on the principle of least privilege and of separation of duty and the permissions are associated with the roles and users associated with these roles. Thus if a users role in the organisation changes only the user association needs to be changed.

B. Operation

The administrator/user logs in through the TLS interface, if the administrator/user is authenticated the user identification number is forwarded to the access control module through the TCP/IP connection. The access control scheme (RBAC) is then applied to the user identification number and the necessary resources and/or services are allocated to the user. The MySQL database maintains all the entities, fields and elements of the RBAC scheme. An administrator only performs operations on the RBAC scheme (updating entities and reviewing activity and error logs). A user interacts with ESD according the user associations in the RBAC scheme. Because the permission associations are performed on data point (Data points are input and output variables in the network) level, it is presumed that the user does have access to the entire data point.

C. Structure

The goal of the access control module is to provide added functionality to the IGUANA gateway, through the successful application of RBAC on the resources and/or services offered by the ESD protocol. The access control module is designed to interface with the existing ESD module, a TLS module and a MySQL server placed on the IGUANA gateway. The access control module is situated on the gateway between the ESD module and the TLS module, as shown in the figure below.

The MySQL server maintains the relational database and tables needed by the access control module. The MySQL server receives, parses, optimise and execute the queries issued by the access control module.

The ESD module is responsible for gathering information from any connected FANDs. The access control module has the responsibility of allocating system resources and services offered by the ESD module.

A TLS module was designed in conjunction with the access control module. The TLS [7] module is responsible authenticating the user (through the use of smart card technology) and implements the SSL (Secure socket layer) protocol to encrypt and decrypt data transferred over the Internet. The TLS is beyond the scopy of this document.

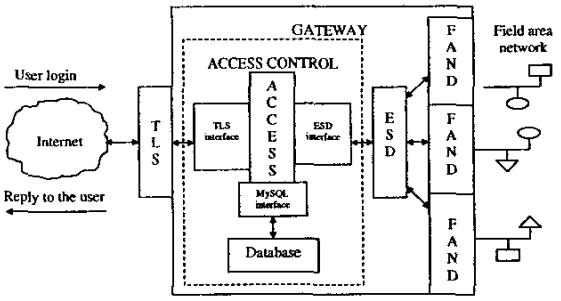

Fig. 6. The figure shows the proposed structure of the IGUANA gateway. Note that the dashed rectangle indicates the placement of the access control module.

D. Elements of the access control module

This section contains a brief description of the sub-modules, interfaces and relational database that forms the access control module.

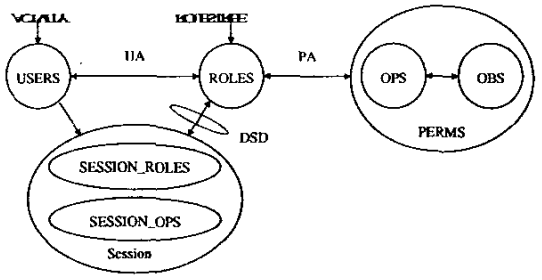

Fig. 7. The figure shows the relationship between the implemented RBAC model elements of the access control module. The double-sided arrows illustrate a many-to-many relationship. The single-sided arrows illustrate a one-to-many relationship.

Administrative sub-module. The administrative sub-module's function is to provide the administrator of the access control module with the ability to create, update, maintain and review the entities of the RBAC model.

The RBAC entities are written in uppercase, because they represent the names of the MySQL tables that store the elements of the specific RBAC entity.

The access control module implements the following RBAC model entities (Figure 7):

Users (USERS): The set of IGUANA gateway users, authenticated by the TLS module, who use the access control module.

Roles (ROLES): The set of named duties or job functions within the IGUANA project.

Sessions (SESSION_OPS and SESSION_ROLES): A mapping between an IGUANA gateway user and an activated subset of authorised roles.

Operations (OPS): The ESD command set, including the event commands regarding the creation, maintenance and deletion of events processed by the ESD event handler.

Objects (OBS): The available nodes, datapoints and events stored on the ESD module.

Permissions (PERMS): A logical association of operations and objects.
User association (UA): An association between an IGUANA gateway user and a role defined by the IGUANA gateway administrator.
Permission association (PA): An association between a role defined by the IGUANA gateway administrator and an active permission.

Role hierarchies (ROLESTREE): The roles are be arranged in the form of a hierarchical structure, thus depicting a roles position within the IGUANA project.

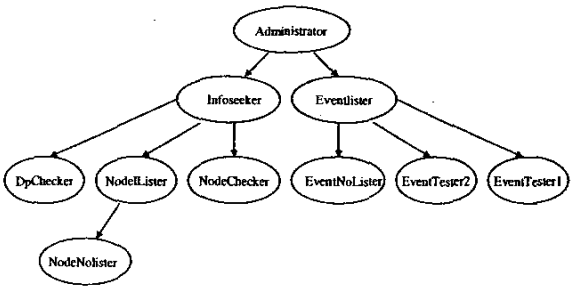

Fig. 8. The figure shows an example of a role hierarchical structure. Note that this hierarchical structure is used to test the hierarchical RBAC functions of the access control module.

The software flow diagram, Figure 9, shows the flow of data during use of the administrative sub-module.

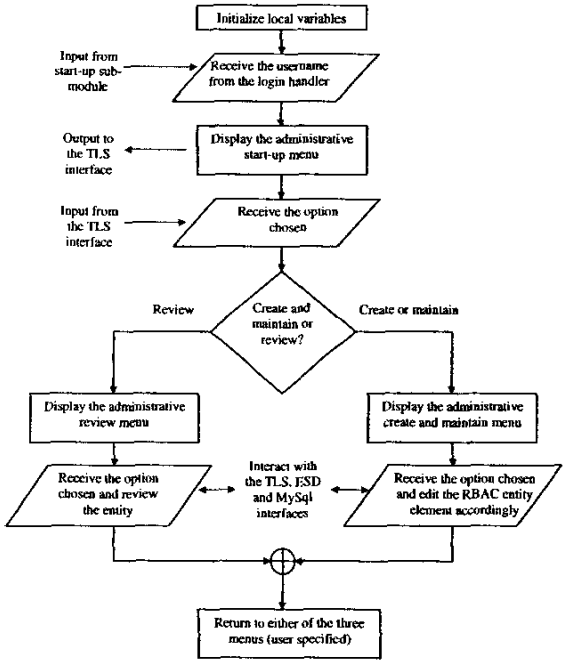

Fig. 9. The software flow diagram shows the primary administrative processes involved.

User sub-module. The user sub-module's function is to provide the user of the access control module the ability to activate or deactivate authorized roles, to perform requested operations and to review authorised permissions (refer to Figure 10). The operations performed are commands issued to the ESD module, which in turn replies with an error, basic reply or an extended response.

E. Interfaces

TLS interface: The TLS interface is implemented as a TCP/IP server. The TLS module authenticates each user that logs onto the IGUANA gateway.

ESD interface: The ESD interface is implemented as a TCP/IP client. The ESD interface connects to the ESD module. The ESD module receives commands from the access control module, performs the operation and responds with the data requested or an acknowledgement.

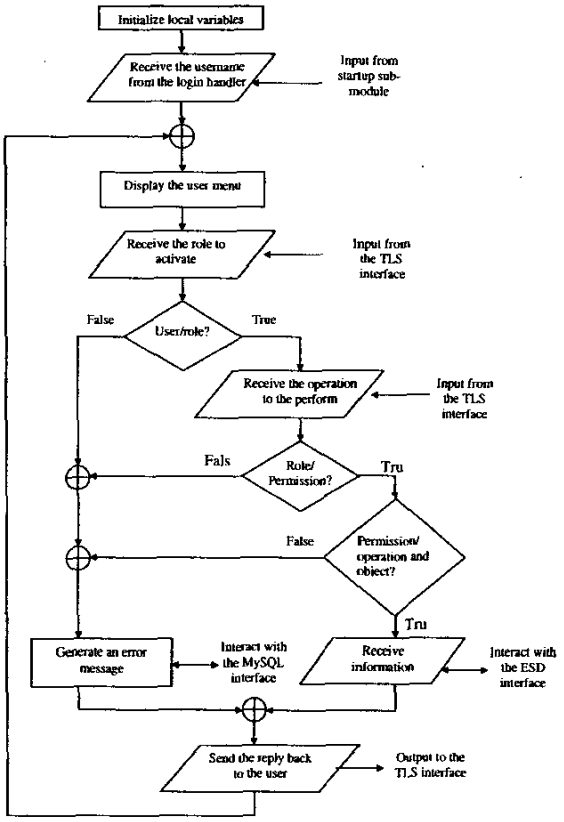

Fig. 10. The software flow diagram shows the primary user processes involved. Note that all decisions are made based on the information contained by the MySQL relational database.

MySQL interface: The MySQL interface is implemented as a TCP/IP client. The MySQL interface connects to the MySQL server. All the sub-modules use the MySQL interface to issue queries to the MySQL server and to return or print the result.

V. RESULTS

The most important results were obtained by performing the following three experiments. The results are not given here due to size constraints.

A. Experiment 1: Testing sets and relations of the DSD

The experiment was designed to prove that the DSD functions of both the administrative and user sub-module work. The administrative sub-module was used in the setup of the DSD sets and relations, and the user sub-module was used to test the application of the DSD functions and rules upon role activation.

The user sub-module successfully enforced the DSD sets.

B. Experiment 2: Testing sets and relations of the role hierarchy

The experiment was designed to prove that the role hierarchical structure (shown in figure 8) can be successfully configured or reviewed by the administrative sub-module, and be successfully applied by the user sub-module. The administrative module was successfully used to configure the role hierarchical structure and the user was, through role activations, able to receive permissions to ESD nodes and data points.

C. Experiment 3: Testing the application of RBAC on generated events

The experiment was designed to prove that the RBAC control policy can successfully be configured by the administrative sub-module and be correctly applied by the user sub-module.

The event command set is the largest and most complex commands in the ESD command sets. The events are broken down into four segments.

- Event criteria: Defining a criterion under which an event is triggered.
- Event action: Defines the operations to perform when the event is triggered.
- Event description: Human readable description of the event.
- Event ICC parameters: Unknown event relevance.

The access control module further segments the four segments into:

- operations identified within each of the ESD segments,
- objects identified within each segment,
- user variables (such as the maximum number of log entries within the ESD), and
- user textual messages.

The creation and deletion of an event causes the access control module to update the role's (used to create the permission) permission associations. The user must be able to maintain and review the newly created event. Note that the permission regarding the event is declared inactive if the event is deleted by an administrator directly connected the ESD. The administrative sub-module successfully configured roles, with permissions, to create, delete and maintain events. The user sub-module was successfully used to create and delete events.

Note the bulk of the response time of the access control module can be attributed to the interaction with the ESD. The experiments showed that network latency and the encryption done by the TLS before the information is received by the user plays a huge role in packet transfer. Nearly 60% of the response time can be attributed to those two factors.

VI. CONCLUSION

The goal was to determine whether the access control module could be successfully implemented and used on the IGUANA gateway. The access control module successfully applies the RBAC policy to all the operations offered by the ESD. The access control module is the beginning of a long line of experimental applications of different access control policies.

Further work includes converting the above implementation into a machine automated system.

REFERENCES

[1] Barkley, J.F. 1997, *Comparing simple role based access control models and access control lists* (Report No. 973-3346), Gaithernurg, M.D.: National Institute of Standards and Technology.

[2] Barkley, J.F., Beznosov, B., Uppal, J., 1999. *Supporting relationships in access control using role based access control.*, IEEE Computer Society Press.

[3] Gallaher, M.P., O'Conner, A.C. and Kropp, B., 2002. *The economic impact of role based access control* (Report No. 07007,012). Gathersburg, M.D.: National Institute of Standards and Technology.

[4] Ferraiolo, D.F., Barkley, J.F. and Khun, D.R., 1999. *A role based access control model and reference implementation withing a corporate intranet.* ACM Trans. Inf. Syst. Sec 2, 1.

[5] Ferraiolo, D.F., Cugini, J.A. and Khun, D.R., 1995. *Role based access control (RBAC): Feature and Motivations.* In: Proceedings of the Annual Computer Society Application Conference: IEEE Press, Los Alamitos, Calif.

[6] Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R., 2001. *Proposed NIST standard for role based access control.* ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, 224-274

[7] Shulze, S., 2002. *Secure Internet communication for the IGUANA gateway.* Final year project report, Project EPR400, Department of Electrical, Electronic and Computer Engineering, University of Pretoria.