A SMART CARD BASED PREPAID ELECTRICITY SYSTEM

M. Wassim Raad[1], *Muhammad Sallout*[2]

[1]Computer Engineering Department,
King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia
mwraad@ccse.kfupm.edu.sa

ABSTRACT

Over the past several years, smart cards have achieved a growing acceptance as a powerful tool for security, identification, and authorization. Financial card issuers are moving to replace magnetic stripe cards with chip cards to reduce counterfeiting and fraud. The increasing computational power placed on the chip along with advances in cryptography has made the smart card a very powerful tool for identification. The advent of multi-application smart card operating systems for both contact and contact less applications has put smart cards on the edge of information technology. This paper introduces a novel 3-tier smart card secure solution for prepaid electricity. The proposed system uses an IP-based controller in addition to a power meter, providing efficient online control of the amount of electricity consumed by the user. The user can use the card to log in to the service provider company, as well as toping up his smart card for additional power needed.

1. INTRODUCTION

In the last decade, smart cards evolved from basic memory cards to complex systems on chips with expanding processing power. This has opened the avenue to many applications such as financial transactions, e-commerce, physical access control, health, and transportation services [1]. The smart card, an intelligent token, is a credit card sized plastic card embedded with an integrated circuit chip. It provides not only memory capacity, but computational capability as well. A smart card usually consists of a ROM or flash memory, EEPROM and a CPU. Access to data stored on the card is under the control of the smart card operating system.

The card operating system not only makes the smart card secure for access control, but can also store a private key for a public key infrastructure system. Lately, the industry has come up with 32-bit smart card processors having more than 400Kbytes of EEPROM, and a memory management and protection unit serving as a hardware firewall. This hardware firewall enables secure separation of adjacent applications, as well as being the basis for secure downloading of applications. The self-containment of smart card makes it resistant to attack as it does not need to depend upon potentially vulnerable external resources. Because of this characteristic, smart cards are often used in several applications which require strong security protection and authentication [2]. In addition to information security, smart cards achieve greater physical security of services and equipments, because a smart card restricts access to all but authorized users.

Furthermore, the smart card can be used as a credit/debit bank card which allows it to be used effectively in e-commerce applications. The multi-application smart card, along with the advent of open platform smart card operating systems, brings the only realistic option for managing multiple electronic transactions nowadays. It is a cost effective

secure way to manage transactions electronically Manufacturers, issuers and users have recognized the value of one card that handles multi-applications. A multi-application card will be able to automatically update new services and existing applications, change and store user profiles for each application and be accepted by a range of devices-PC, POS, mobile phones [3]. One of the most valuable applications is in using the smart card to buy energy. Domestic consumers could for instance buy energy, at a price based on their previous consumption pattern, from any supplier wherever and whenever they choose. When the customer wants to top up their gas or electricity credit they visit a vending machine which uses the consumption data stored on their card to allocate a tariff and calculates how much energy to offer the consumer for their money [4]. Recently, the portal technology has been playing an increasing role in computing. Service providers are rolling out portals to allow users to create customized web sites that display exactly the information of interest. Corporations are rolling out portals to provide employees and business partner's customizable access to corporate information, including news feeds from external providers, or email, calendar and access to billing system, in addition to other web services. For web enabled energy services, and with the advent of home networking technology, power companies and service providers can provide value-added services delivered to the homes, like energy management, to generate additional revenue as well as to increase convenience and loyalty.

In this paper, we propose a novel and simple prototype of a web enabled smart card based solution for controlling the consumption of electricity in a home environment [5].

2. LOGICAL FILE STRUCTURE OF A TYPICAL SMART CARD

In terms of data storage, a smart card is organized in a hierarchical form through directories. Similar to MS-DOS, there is one master file (MF) which is like the root directory. Under the root, we can have different files which are called elementary files (EFs). We can also have various subdirectories called dedicated files (DFs). Every DF is dedicated for a separate application. Under each subdirectory will be elementary files again. The master file is implicitly selected after the smart card is reset. The useful data that are needed for an application are located in the (EFs). EFs may be placed directly under the MF or under a DF. Since the MF is a special sort of DF, it goes without saying that in a single application smart card, all application files can be placed directly under the MF. Additional DFs can be placed within an application DF. For example, a DF placed directly under the MF could be dedicated to the 'Corporate ID' application. An additional level of DFs within the application DF could contain the files for the languages supported, such as 'English' and 'Arabic'. Figure 1 shows logical view of a smart card file structure.

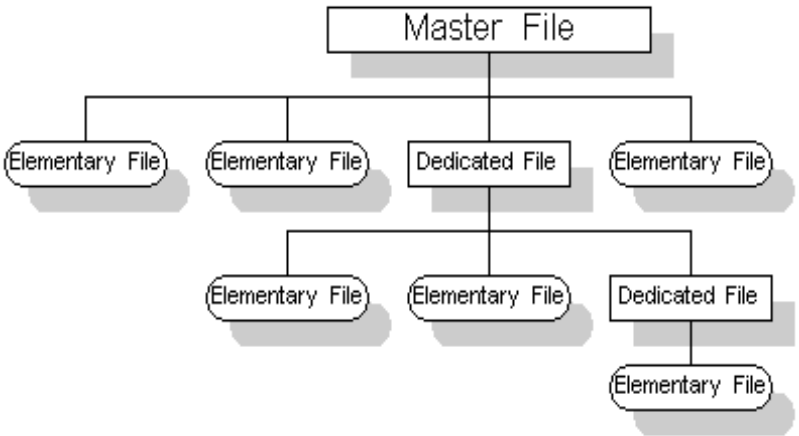

Figure 1. Logical file structure of smart card

The elementary file (EF) simply consists of a header and the body which stores the data. The header of the file stores information about the file such as identification number, description, types, size, and so on. Particularly, it stores the attribute of the file which states the access conditions and current status. Access of the data in the file depends on whether those conditions can be fulfilled or not.

In short, the file structure of the smart card operating system is similar to other common operating systems such as MS-DOS and UNIX. However, in order to provide greater security control, the attribute of each file is enhanced by adding access conditions and file status fields in the file header. Moreover, file lock is also provided to prevent unauthorized access. This security mechanism provides a logical protection of the smart card [6, 7].

3. PREPAID ELECTRICITY

Since the last decades of the past century, scientists, researchers and public people have been worried about energy conservation. People spend much more power than what they actually need and that results in a huge loss of energy. Moreover, the continuous increase in the universal energy prices has resulted in a huge economical loss. Thus we are proposing a prepaid electricity smart card based system so people can buy specific amount of energy to use it only when then need. People can register for this service and charge their accounts through the Internet. The proposed system is based on an IP-based controller called TINY, and a WATTNODE type power meter which interrupts the controller at a regular interval based on the consumption of electricity to update the balance based on a certain tariff. The power meter we used, interrupts the controller at a

rate of 0.75Wph, so based on the particular tariff used and the amount of power consumption needed, the correct amount of money to be loaded into the card can be easily calculated and programmed into the chip. The unique feature about this system is that the electric utility in the home environment can be accessed remotely from the supplier server due to the fact that the controller is IP-based, without the need for a PC on site, which reduces the cost of the system drastically. People now can buy electricity in advance, using the so-called prepaid electricity cards. The proposed prepaid smart card can also be used to manage electricity consumption in a hotel room, as well as accessing the room itself.Thus, people can consume only as much power as they really need. The main role of the smart card is summarized in two things:

- Authenticating the user or log in
- Updating the balance in the card based on the given tariff and the electricity consumption profile of the user stored in the smart card. See Figure 2 for the proposed system.

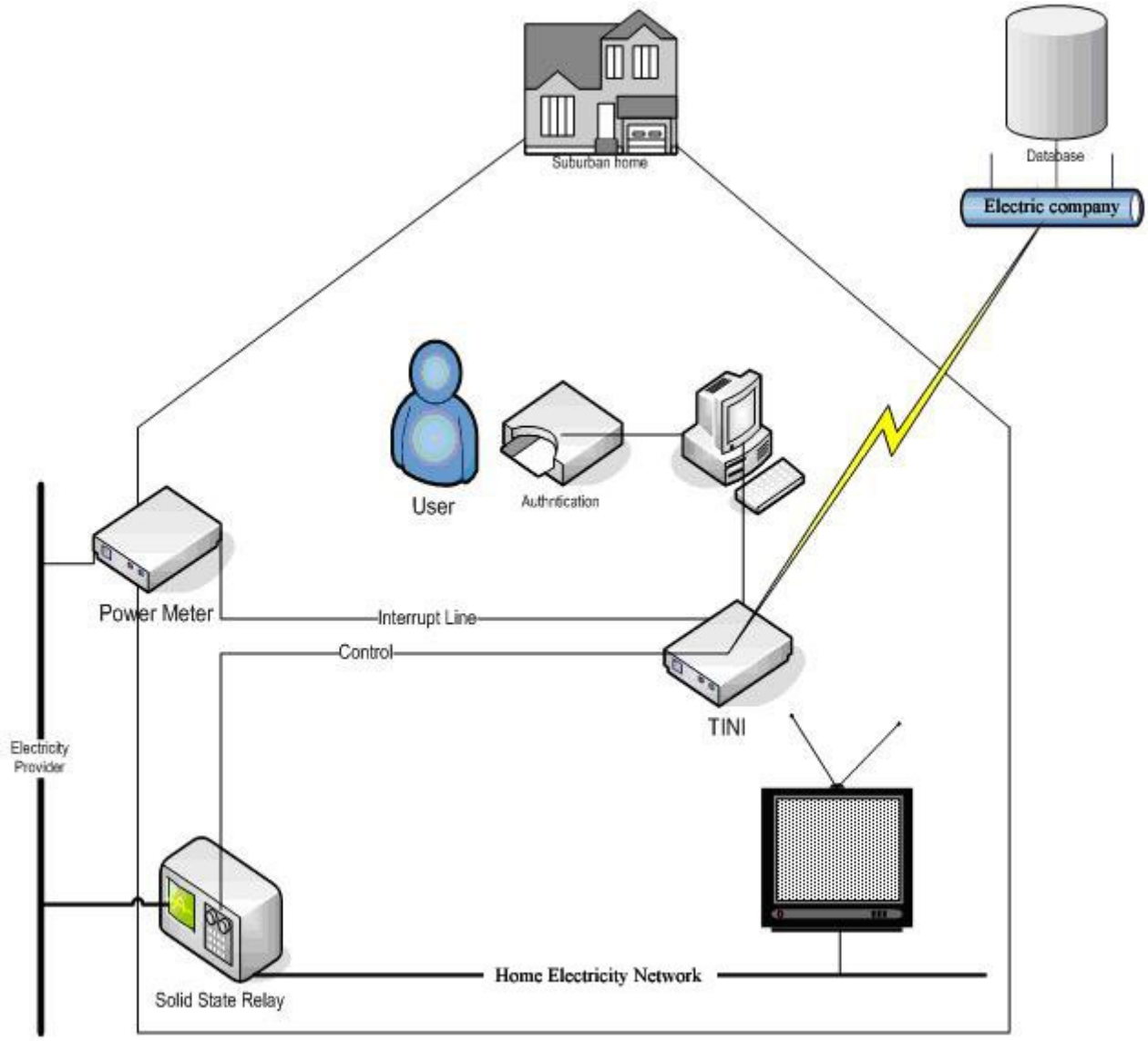

Figure 2 Prepaid Electricity System Design

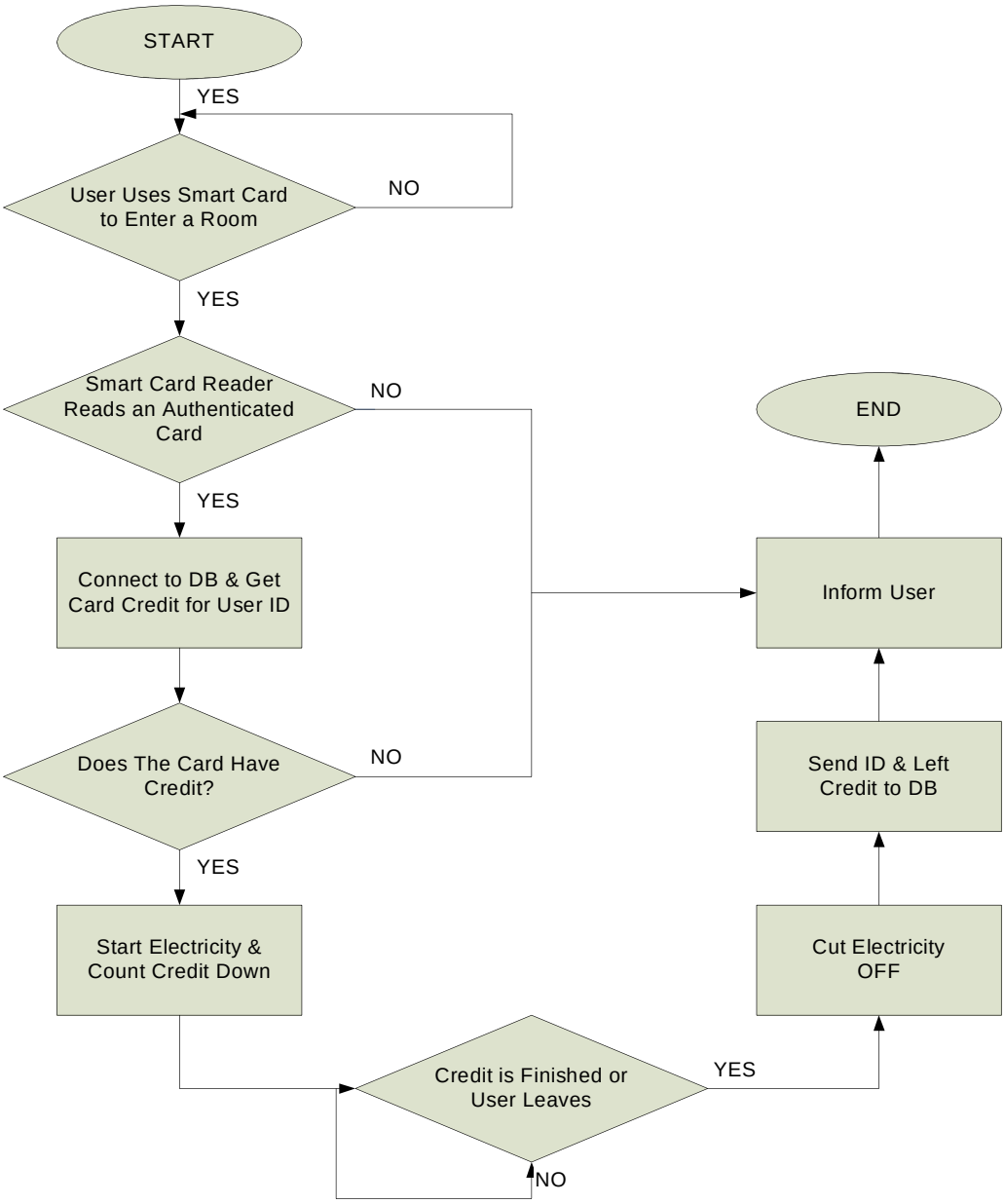

Figure 3. Flow Chart for Prepaid Electricity

As shown in Figure2, the main role of the TINY IP-based controller is to control the flow of electricity using a solid state relay, based on the consumption profile of the user. The TINY controller is programmed to determine the exact amount of electricity measured by the power meter by counting the exact number of interrupts corresponding to the amount of electricity which the user intend to consume. The controller also provides remote access to the electricity supplying company server. We used a PC/SC compatible smart card reader with ACOS1-8K smart card. The smart card here is used as a secure token to log in to the data base of the company server, in a client server manner, for controlling consumption of electricity as well as toping up the smart card with additional money [8]. See Figure3 for the authentication and the online purchasing protocol of the smart card based prepaid electricity system.

4. Proposed Architecture

In this section, the overall design of the system is discussed in terms of two approaches: two-tier architecture, and three-tier architecture [8].

Both two and three tier architectures are examples and variations of the well-know client/server computing model. The model was proposed as an alternative to centralized mainframe and time sharing computing. In this model, the client interacts with the user possibly via a GUI interface, and requests on-line services from the server. The server, on the other hand, answers these requests and provides the services.

The fact that this system is spread across more than two different entities suggests three-tier architecture. Such architecture brings clear logical structure to the system. A major advantage of the three-tier architecture over the two-tier architecture is scalability. Three-tier architecture supports hundreds of users while the two-tier architecture is known to work well with less than 100 users. This is because the latter maintains connections with each client while the three-tier architecture is able to manage these connections (via queuing, for example) better due to the middle tier. Scalability is a major requirement for our design since there could be thousands of customers who are trying to update the balance on their prepaid electricity card and trying to establish a session with the servers. This requirement by itself is sufficient to justify the three-tier architecture over the two-tier architecture. See Figures 4 and 5 for the two-tier and three tier architectures respectively.

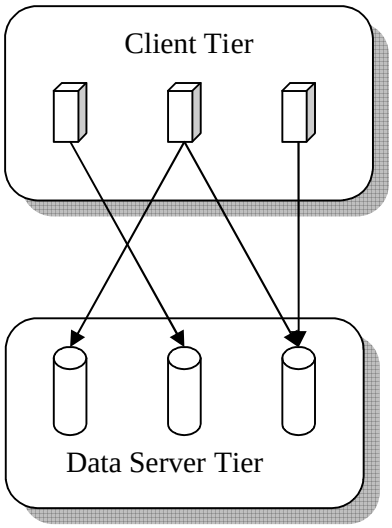

Figure4: Two-Tier Architecture

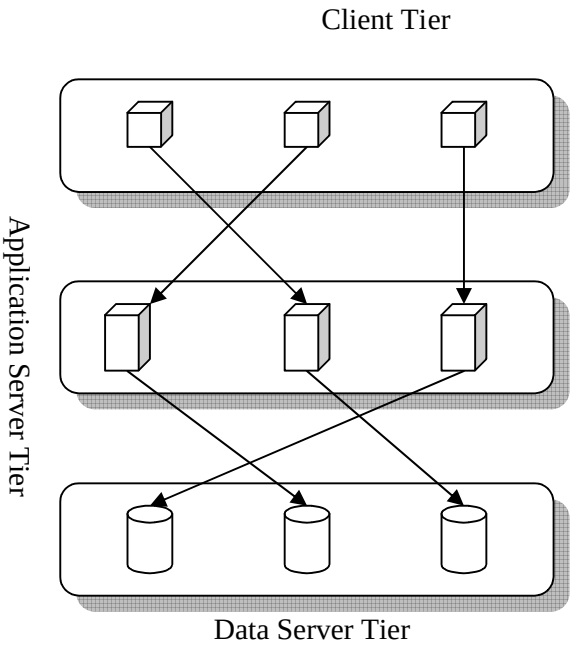

Figure 5. Three Tier architecture

In addition, with the three-tier architecture most of the application (business) logic like locating the appropriate database, checking authority, generating query, is moved to the middle tier. Therefore, in the case of the three-tier architecture, changes in the business logic result in less client tier changes. Another advantage of the three-tier architecture is that its data security is increased because the client tier no longer can access the data directly; it has to go through the middle tier first [9,10].

However, the three-tier architecture has its own disadvantages. Designing and developing three-tier architecture software is a complex process, and some of its available technologies are complex too (for

example, J2EE technology or .NET technology). In addition, separation of business logic from presentation logic is not always clear.

Figure 6 shows the main components of the smart card system. The overall system was tested successfully in the lab using a 1000W heater.

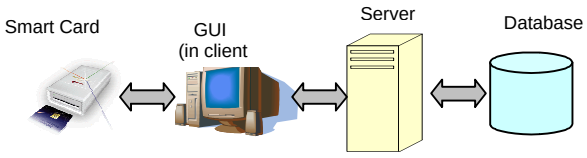

Figure 6. Main components of the smart card system

5. CONCLUSION

A secure smart card based system for e-payment, implemented on prepaid electricity over the internet, was proposed. The smart card system has been designed and implemented successfully using a three tier model client-server system, which was shown to be superior over the two tier client-server system model. The proposed system has the benefit of using a secure smart card to log in to the network, and control the amount of money needed to be spent for the required electricity consumption based on the user profile stored on the card. The proposed system has also the unique feature of using an IP-based controller which provides remote access to company server without the additional cost of a PC. Work is ongoing for shifting to a true open platform multiplication smart card environment.

6. REFERENCES

[1] Jean-Francois Dhem and Nathalie Feyt, IEEE Micro, Vol:21, issue:6, Nov-Dec 2001, p.14-25.

[2] Dirk Huseman, Concurrency IEEE, Vol7, issue2, April-June 1999, pp.24-27.

[3] Multi-functionality for smart cards, www.oberthurusa.com/whitepapers-multi.asp

[4] John Cowburn, IEE Review, Vol:47, issue 4, July 2001.

[5] Uwe Hansmann, Lothar Merk, Martin S. Nicklous, and Thomas Stober, Pervasive Computing, Springer-Verlag Berlin Heidelberg New York, 2003.

[6] W. Ranki and W. Effing, Smart Card Handbook, John Wiley and Sons 2000.

[7] Mike Hendry, Smart Card Security and Applications, ARTECH HOUSE INC. 1997.

[8] Client/Server Software Architectures: An Overview, Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/str/descriptions/clientserver_body.html

[9] Two Tier Software Architectures, Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/str/descriptions/twotier.html#512860

[10] Three Tier Software Architectures, Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/str/descriptions/threetier.html#34492