Smart card system

The term “ client/server” implies that clients and servers are separate logical entities that work together, usually over a network, to accomplish a task. Client/server is more than a client and a server communicating across a network.

Client/server uses asynchronous and synchronous messaging techniques with the assistance of middle-ware to communicate across a network. Client/Server uses this approach of a client (UI) and the server (database I/O) to provide its robust distributed capabilities. The company, Sigma has used this technique for over 15 years to allow its products to be ported to multiple platforms, databases, and transaction processors while retaining a product’s marketability and enhanced functionality from decade to decade. Sigma’s client/server product uses an asynchronous approach of sending a message to request an action and receives a message containing the information requested. This approach allows the product to send intensive CPU processing requests to the server to perform and return the results to the client when finished. Sigma’s architecture is based on re-usability and portability.

Sigma currently uses a standard I/O routine, which is mutually exclusive from the user interface. Sigma’s current architecture supports character-based screens and a variety of databases where the user interface is independent of the database access. This architecture corresponds directly to the architecture used in a GUI Client/Server environment. Sigma’s client/server product uses an asynchronous approach of sending a message to request an action and receives a message containing the information requested. A traditional client/server application is that of the File Server where clients

request files from the File Server. This results in the entire file being sent to the client but necessitates many message exchanges across the network.

Another traditional client/server application is that of the Database Server where clients pass SQL requests to the server. The Database Server executes each SQL statement and passes the results back to the client. Open Database Connectivity (ODBC) is often used by a client to send SQL requests to the server to process. ODBC provides a standard SQL interface for sending requests to the server. The Remote Procedure Call (RPC) is an extension of the traditional client/server model suited to transaction processing environments. It allows for the creation of a Transaction Server.

Clients call a remote procedure and pass parameters to it. A single message allows the Transaction Server to execute stored (compiled) database statements and return the results to the client. This distribution of processing reduces network traffic and improves performance. Site autonomy can also be increased by limiting database modifications to locally executing applications. Remote Procedure Call (RPC). The Remote Procedure Call (RPC) is a mechanism that allows programs to communicate with each other.

1. 2 IP protocolTCP sends each datagrams to IP. Of course it has to tell IP the Internet address of the computer at the other end. This is all IP is concerned about. It doesn’t care about what is in the datagram, or even in the TCP header. IP’s job is simply to find a route for the datagram and get it to the other end.

In order to allow gateways or other intermediate systems to forward the datagram, it adds its own header. The main things in this header are the source and destination Internet address (32-bit addresses, like 128. 6. 4. 194), the protocol number, and another checksum.

The source Internet address is simply the address of source machine. The destination Internet address is the address of the other machine. The protocol number tells IP at the other end to send the datagram to TCP. Although most IP traffic uses TCP, there are other protocols that can use IP, so IP have to been told which protocol to send the datagram to. Finally, the checksum allows IP at the other end to verify that the header wasn’t damaged in transmit.

TCP and IP have separate checksums. IP has to be able to verify that the header didn’t get damaged in transmit, or it could send a message to the wrong place. It is both more efficient and safer to have TCP compute a separate checksum for the TCP header and data. IP addresses are used to deliver packets of data across a network and have what is termed end-to-end significance.

This means that the source and destination IP address remains constant as the packet traverses a network. Each time a packet travels through a router, the router will reference it’s routing table to see if it can match the network number of the destination IP address with an entry in its routing table. If a match is found, the packet is forwarded to the next hop router for the destination network in question. If a match is not found, then the packet may be forwarded to the router defined as the default gateway, or the router may

drop the packet. Packets are forwarded to a default router in the belief that the default router has more network information in its routing table and will therefore be able to route the packet correctly on to its final destination.

This is typically used when connecting a LAN with PCs on it to the Internet. Each PC will have the router that connects the LAN to the Internet defined as its default gateway. A default gateway is seen in a routing table of a host as follows: the default route 0. 0. 0.

0 will be listed as the destination network, and the IP address of the default gateway will be listed as the next hop router. If the source and destination IP addresses remain constant as the packet works its way through the network, how is the next hop router addressed? In a LAN environment this is handled by the MAC (Media Access Control) address. The key point is that the MAC addresses will change every time a packet travels though a router, however, the IP addresses will remain constant. Subnet masks are essential tools in network design, but can make things more difficult to understand. Subnet masks are used to split a network into a collection of smaller subnetworks.

This may be done to reduce network traffic on each subnetwork, or to make the internetwork more manageable as a whole. 1. 3 Network Protocol in LANMost LANs connect workstations and personal computers. Each node (individual computer) in a LAN has its own CPU with which it executes programs, but it is also able to access data and devices anywhere on the LAN. This means that many users can share expensive devices, such as laser printers, as well as data. Users can also use the LAN to communicate with each other, by sending e-mail or engaging in chat sessions.

The following characteristics differentiate one LAN from another:* Topology: The geometric arrangement of devices on the network. For example, devices can be arranged in a ring or in a straight line.* Protocols: The rules and encoding specifications for sending data. The protocols also determine whether the network uses peer-to-peer or client/server architecture.

* Media: Devices can be connected by twisted-pair wire, coaxial cables, or fiber optic cables. Some networks do without connecting media altogether, communicating instead via radio waves. 1. 3. 1. File Transfer ProtocolThe File Transfer Protocol (FTP) provides the basic elements of file sharing between hosts.

FTP uses TCP to create a virtual connection for control information and then creates a separate TCP connection for data transfers. The control connection uses an image of the TELNET protocol to exchange commands and messages between hosts. 1. 3. 2.

User Datagram ProtocolThe User Datagram Protocol (UDP) provides a simple, but unreliable message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts. 1. 3.

3. Transmission Control ProtocolTCP provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgement with retransmission of packets when necessary. TCP uses a 32-bit sequence number that counts bytes in the data stream. Each TCP packet contains the starting sequence number of the data in that packet,

and the sequence number (called the acknowledgment number) of the last byte received from the remote peer. With this information, a sliding-window protocol is implemented. Forward and reverse sequence numbers are completely independent, and each TCP peer must track both its own sequence numbering and the numbering being used by the remote peer.

1. 4 Winsock 2. 0 ArchitectureWindows Sockets version 2. 0 (WinSock 2) formalizes the API for a number of other protocol suites– ATM, IPX/SPX, and DECnet–and allows them to coexist simultaneously. It still retains full backward compatibility with the existing 1.

1–some of which is clarified further–so all existing WinSock applications can continue to run without modification (the only exception are WinSock 1. 1 applications that use blocking hooks, in which case they need to be re-written to work without them). WinSock 2 goes beyond simply allowing the coexistence of multiple protocol stacks, in theory it even allows the creation of applications that are network protocol independent. A WinSock 2 application can transparently select a protocol based on its service needs. The application can adapt to differences in network names and addresses using the mechanisms WinSock 2 provides.

1. 4. 1 WinSock 2 ArchitectureWinSock 2 has an all-new architecture that provides much more flexibility. The new WinSock 2 architecture allows for simultaneous support of multiple protocol stacks, interfaces, and service providers. There is still one DLL on top, but there is another layer below, and a standard service provider interface, both of which add flexibility.

WinSock 2 adopts the Windows Open Systems Architecture (WOSA) model, which separates the API from the protocol service provider. In this model the WinSock DLL provides the standard API, and each vendor installs its own service provider layer underneath. The API layer “ talks” to a service provider via a standardized Service Provider Interface (SPI), and it is capable of multiplexing between multiple service providers simultaneously. 1.

5 Transmission Control Protocol (TCP)Initially, TCP was designed to recover from node or line failures where the network propagates routing table changes to all router nodes. Since the update takes some time, TCP is slow to initiate recovery. The TCP algorithms are not tuned to optimally handle packet loss due to traffic congestion. Instead, the traditional Internet response to traffic problems has been to increase the speed of lines and equipment in order to say ahead of growth in demand.

TCP treats the data as a stream of bytes. It logically assigns a sequence number to each byte. The TCP packet has a header that says, in effect, “ This packet starts with byte 379642 and contains 200 bytes of data.” The receiver can detect missing or incorrectly sequenced packets. TCP acknowledges data that has been received and retransmits data that has been lost.

The TCP design means that error recovery is done end-to-end between the Client and Server machine. There is no formal standard for tracking problems in the middle of the network, though each network has adopted some ad hoc tools. To insure that all types of systems from all vendors can communicate, TCP/IP is absolutely standardized on the LAN. However, larger networks

based on long distances and phone lines are more volatile. New technologies arise and become obsolete within a few years.

With cable TV and phone companies competing to build the National Information Superhighway, no single standard can govern citywide, nationwide, or worldwide communications. The original design of TCP/IP as a Network of Networks fits nicely within the current technological uncertainty. TCP/IP data can be sent across a LAN, or it can be carried within an internal corporate SNA network, or it can piggyback on the cable TV service. Furthermore, machines connected to any of these networks can communicate to any other network through gateways supplied by the network vendor. 1.

6. Data packet transmissionData packet transmission consists of a series of handshaking sequences where the sending side of the end node/repeater local port, point-to-point connection makes a request and the other side acknowledges the request. The sequence of sending a data packet transmission is requested be an end node and controlled by the repeater. When a data packet transmission is about to occur: If an end node has a data packet ready to send, it transmits either a Request_Normal or Request_high control signal. Otherwise, the end node transmits the Idle_Up control signal.

1. The repeater polls all local ports to determine which end nodes are requesting to send a data packet and at what priority level that request is (normal or high). 2. The repeater selects the next end node with a high priority request pending.

Ports are selected in port order. If not high priority requests are pending, then the next normal priority port is selected (in port order). This selection causes the selected port to receive the Grant signal. Packet transmission begins when the end node detects the Grant signal. 3. The repeater then sends the Incoming signal to all other end nodes, alerting them to the possibility of an incoming packet.

The repeater decodes the destination address from the frame being transmitted as it is being received. 4. When an end node receives the Incoming control signal, it prepares to receive a packet by stopping the transmission of requests and listening on the media for the data packet. 5. Once the repeater has decoded the destination address, the packet is delivered to the addressed end node or end nodes and to any promiscuous nodes. Those nodes not receiving the data packet receive the Idle_Down signal from the repeater.

6. When the end node(s) receive the data packet, they return to their state prior to the reception o f the data packet, either sending an Idle_Up signal or making a request to send a data packet. 1. 7. ConclusionWinSock 2 has an all-new architecture that provides much more flexibility.

The new WinSock 2 architecture allows for simultaneous support of multiple protocol stacks, interfaces, and service providers. It is suitable for win32 platform however, it is designed back compatible that is mean even the win95 also can use it without conflict. The 32-bit wsock32. dll ships with Windows NT and Windows 95 and runs over the Microsoft TCP/IP stack. These 32-bit environments also have a winsock.

dll file that acts as a “ thunk-layer” to allow 16-bit WinSock applications to run over the 32-bit wsock32. dll. Conversely, Microsoft’s Win32s installs a 32-bit wsock32. dll thunk layer in 16-bit Windows environments (Windows version 3.

1 and Windows for Workgroups 3. 11) over any vendor’s WinSock DLL currently in use. LANs are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line; but the distances are limited, and there is also a limit on the number of computers that can be attached to a single LAN. File Transfer Protocol (FTP), User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) are commonly applies in LANs’ transactions. Data transmit using UDP is faster than TCP but TCP has better data security and data integrity compare to UDP.

The client server architecture basically consists of client machines and server machines. There are 2 form of client server architecture. The first one is that of the File Server where clients request files from the File Server. This results in the entire file being sent to the client but necessitates many message exchanges across the network.

Another traditional client/server application is that of the Database Server where clients pass SQL requests to the server. The Database Server executes each SQL statement and passes the results back to the client. The source and destination IP address remains constant as the packet traverses a network. The packet may be forwarded to the router defined as the default gateway, or the router may drop the packet. Packets are forwarded to a

default router in the belief that the default router has more network information in its routing table and will therefore be able to route the packet correctly on to its final destination.

Chapter 2: Data Encryption and Cryptography Technology2. 1 Introduction To Encryption And Cryptography TechnologyEncryption is the conversion of a piece of data or plaintext into a form, called a ciphertext, which cannot be easily understood by unauthorized people. Meanwhile, decryption is the process of converting encrypted data or ciphertext back into its original form so that it can be understood. The conversion of plaintext into ciphertext or vice versa, should be applied accompany with a cryptographic algorithm. Most encryption algorithm are based on the concept of complex mathematics that work in only one direction and are generally based on the difficulties of factoring verly large numbers (keys) that are used for the encryption.

These large numbers are the products of large prime numbers. Many encryption programs use one key for both encrypting and decrypting messages, which this is known as symmetric cryptography. This is a fast and simple method of encrypting messages and folders and is best for protecting local messages, files and folders. Cryptography is the science of information security.

Examples of cryptography techniques include microdots, merging words with images and etc. However, cryptography is most often associated with scrambling plaintext into ciphertext, then back again. Individuals who practice this field are known as cryptographers. Cryptography mainly

consists of four objectives:* Confidentially – The information cannot be understood by anyone.

* Integrity – The information cannot be altered in storage or transit between sender and receiver.* No-repudiation – The creator or sender of the information cannot deny his intentions in the creation or transmission of the information.* Authentication- The sender and receiver can identify each other’s identity and the origin or destination of the information. 2. 2 DES (Data Encryption Standard) and ImplementationDES is the U. S.

Government’s Data Encryption Standard, a product cipher that operates on 64-bit blocks of data, using a 56-bit key. Triple DES is a product cipher, which like DES, operates on 64-bit data blocks. There are several forms, each of which uses the DES cipher 3 times. Some forms use two 56-bit keys and some use three. The DES “ modes of operation” may also be used with triple-DES. Some people refer to E(K1, D(K2, E(K1, x))) as triple-DES.

This method is intended for use in encrypting DES keys and IVs for “ Automated Key Distribution”. Its formal name is “ Encryption and Decryption of a Single Key by a KeyPair”. Others use the term “ triple-DES” for E(K1, D(K2, E(K3, x))) or E(K1, E(K2, E(K3, x))). Key encrypting keys may be a single DEA key or a DEA key pair. Key pairs should be used where additional security is needed (e.

g., the data protected by the key(s) has a long security life). A key pair shall not be encrypted or decrypted using a single key. Privacy protection using symmetric algorithm DES (the government-sponsored Data Encryption

Standard) is relatively easy in small networks, requiring the exchange of secret encryption keys among each party.

As a network proliferates, the secure exchange of secret keys becomes increasingly expensive and unwieldy. Consequently, this solution alone is impractical for even moderately large networks. DES has an additional drawback, it requires sharing of a secret key. Each person must trust the other to guard the pair’s secret key, and reveal it to no one. Since the user must have a different key for every person they communicate with, they must trust each and every person with one of their secret keys.

This means that in practical implementations, secure communication can only take place between people with some kind of prior relationship, be it personal or professional. Fundamental issues that are not addressed by DES are authentication and non-repudiation. Shared secret keys prevent either party from proving what the other may have done. Either can surreptitiously modify data and be assured that a third party would be unable to identify the culprit. The same key that makes it possible to communicate securely could be used to create forgeries in the other user’s name. 2.

3 RSA-based Cryptographic SchemesThe RSA algorithm was invented by Ronald L. Rivest, Adi Shamir, and Leonard Adleman in 1977. There are a variety of different cryptographic schemes and protocols based on the RSA algorithm in products all over the world. The RSAES-OAEP encryption scheme and the RSASSA-PSS signature scheme with appendix is recommended for new applications. 2.

3. 1 RSAES-OAEP (RSA Encryption Scheme – Optimal Asymmetric Encryption Padding)It is a public-key encryption scheme combining the RSA algorithm with the OAEP method. The inventors of OAEP are Mihir Bellare and Phillip Rogaway, with enhancements by Don B. Johnson and Stephen M. Matyas. 2.

3. 2 RSASSA-PSS (RSA Signature Scheme with Appendix – ProbabilisticSignature Scheme)It is an asymmetric signature scheme with appendix combining the RSA algorithm with the PSS encoding method. The inventors of the PSS encoding method are Mihir Bellare and Phillip Rogaway. During efforts to adopt RSASSA-PSS into the P1363a standards effort, certain adaptations to the original version of RSA-PSS were made by Bellare and Rogaway and also by Burt Kaliski (the editor of IEEE P1363a) to facilitate implementation and integration into existing protocols.

Here is a small example of RSA Plaintexts are positive integers up to 2^{512}. Keys are quadruples (p, q, e, d), with p a 256-bit prime number, q a 258-bit prime number, and d and e large numbers with (de – 1) divisible by (p-1)(q-1). We define E_K(P) = P^e mod pq, D_K(C) = C^d mod pq. All quantities are readily computed from classic and modern number theoretic algorithms (Euclid’s algorithm for computing the greatest common divisor yields an algorithm for the former, and historically newly explored computational approaches to finding large `probable’ primes, such as the Fermat test, provide the latter.)Now E_K is easily computed from the pair (pq, e)—but, as far as anyone knows, there is no easy way to compute D_K from the pair (pq, e). So whoever generates K can publish (pq, e).

Anyone can send a secret message to him; he is the only one who can read the messages. The primary advantage of RSA public-key cryptography is increased security and convenience. Private keys never need to transmitted or revealed to anyone. In a secret-key system, by contrast, the secret keys must be transmitted (either manually or through a communication channel), and there may be a chance that an enemy can discover the secret keys during their transmission. 2.

4 JavaTM Cryptography ArchitectureThe Java Security API is a new Java core API, built around the java. security package (and its subpackages). This API is designed to allow developers to incorporate both low-level and high-level security functionality into their Java applications. The first release of Java Security in JDK 1.

1 contains a subset of this functionality, including APIs for digital signatures and message digests. In addition, there are abstract interfaces for key management, certificate management and access control. Specific APIs to support X. 509 v3 certificates and other certificate formats, and richer functionality in the area of access control, will follow in subsequent JDK releases. The Java Cryptography Extension (JCE) extends the JCA API to include encryption and key exchange. Together, it and the JCA provide a complete, platform-independent cryptography API.

The JCE will be provided in a separate release because it is not currently exportable outside the United StatesThe Java Cryptography Architecture (JCA) was designed around these principles:* Implementation independence and interoperability* Algorithm independence and

extensibilityImplementation independence and algorithm independence are complementary: their aim is to let users of the API utilize cryptographic concepts, such as digital signatures and message digests, without concern for the implementations or even the algorithms being used to implement these concepts. When complete algorithm-independence is not possible, the JCA provides developers with standardized algorithm-specific APIs. When implementation-independence is not desirable, the JCA lets developers indicate the specific implementations they require. 2. 5 JavaTM Cryptography Extension (JCE) 1.

2. 1The JavaTM Cryptography Extension (JCE) 1. 2. 1 is a package that provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms. Support for encryption includes symmetric, asymmetric, block, and stream ciphers. The software also supports secure streams and sealed objects.

JCE 1. 2. 1 is designed so that other qualified cryptography libraries can be plugged in as service providers, and new algorithms can be added seamlessly. (Qualified providers include those approved for export and those certified for domestic use only.

Qualified providers are signed by a trusted entity.) JCE 1. 2. 1 supplements the JavaTM 2 platform, which already includes interfaces and implementations of digital signatures and message digests.

This release of JCE is a non-commercial reference implementation that demonstrates a working example of the JCE 1. 2. 1 APIs. A reference

implementation is similar to a proof-of-concept implementation of an API specification.

It is used to demonstrate that the specification is implementable and that various compatibility tests can be written against it. A non-commercial implementation typically lacks the overall completeness of a commercial-grade product. While the implementation meets the API specification, it will be lacking things such as a fully-featured toolkit, sophisticated debugging tools, commercial-grade documentation and regular maintenance updates. The Java 2 platform already has implementations and interfaces for digital signatures and message digests.

JCE 1. 2 was created to extend the Java Cryptography Architecture (JCA) APIs available in the Java 2 platform to include APIs and implementations for cryptographic services that were subjected to U. S. export control regulations. JCE 1. 2 was released separately as an extension to the Java 2 platform, in accordance with U.

S. export control regulations. Important Features of JCE 1. 2. 1* Pure Java implementation.* Pluggable framework architecture that enables only qualified providers to be plugged in.

* Exportable (in binary form only).* Single distribution of the JCE 1. 2. 1 software from Sun Microsystems for both domestic and global users, with jurisdiction policy files that specify that there are no restrictions on cryptographic strengths. 2.

6 ConclusionsDES is available in software as content-encryption algorithm. Several people have made DES code available via ftp. Stig Ostholm [FTPSO]; BSD [FTPBK]; Eric Young [FTPEY]; Dennis Furguson [FTPDF]; Mark Riordan [FTPMR]; Phil Karn [FTPPK]. A Pascal listing of DES is also given in Patterson [PAT87]. Antti Louko <

fi> has written a version of DES with BigNum packages in [FTPAL]. Therefore, we are able to get the DES algorithm and use it in encrypt our administrator password, user passwords and server's database. RSA algorithm is a very popular encryption algorithm. There are collections of links to RSA-related documents on Internet. There are a variety of different cryptographic schemes and protocols based on the RSA algorithm in products all over the world.

The most recommended RSA algorithm is RSAES-OAEP encryption scheme and the RSASSA-PSS signature scheme. We will look into their encoding method and find out a suitable choice for our server's database encryption. The " Java Cryptography Architecture" (JCA) refers to the framework for accessing and developing cryptographic functionality for the Java Platform. It encompasses the parts of the JDK 1. 1 Java Security API related to cryptography (currently, nearly the entire API), as well as a set of conventions and specifications provided in this document. It introduces a " provider" architecture that allows for multiple and interoperable cryptography implementations.

The JavaTM Cryptography Extension (JCE) 1. 2. 1 is used in providing better encryption to system. This package requires JavaTM 2 SDK v 1. 2.

1 or later or JavaTM 2 Runtime Environment v 1. 2. 1 or later already installed. It is an extension for the Java Cryptography Architecture (JCA) APIs available in the Java 2 platform. In our project topic, Control Access of lab computer in a client server environment needs encryption for administrator database and smart card as well.

Therefore, we studying JCE 1. 2. 1 and make use of its cryptographic strengths. Chapter 3: Smart Card Technology3. 1. About The Java Card TechnologyIn the Java Card specifications enable JavaTM technology to run on smart cards and other devices with limited memory.

The Java Card API also allows applications written for one smart card platform enabled with Java Card technology to run on any other such platform. From this two new technology, Java smart card technology becomes more efficient to be used. The Java Card Application Environment (JCAE) is licensed on an OEM-basis to smart card manufacturers, representing more than 90 percent of the worldwide smart card manufacturing capacity. There are several unique benefits of the Java Card technology, such as:* Platform Independent – Java Card technology applets that comply with the Java Card API specification will run on cards developed using the JCAE – allowing developers to use the same Java Card technology applet to run on different vendors’ cards.* Multi-Application Capable – Multiple applications can run on a single card. In the Java programming language, the inherent design around small, downloadable code elements makes it easy to securely run multiple applications on a single card.

* Post-Issuance of Applications – The installation of applications, after the card has been issued, provides card issuers with the ability to dynamically respond to their customer’s changing needs. For example, if a customer decides to change the frequent flyer program associated with the card, the card issuer can make this change, without having to issue a new card.* Flexible – The Object-Oriented methodology of the Java Card technology provides flexibility in programming smart cards.* Compatible with Existing Smart Card Standards – The Java Card API is compatible with formal international standards, such as, ISO7816, and industry-specific standards, such as, Europay/Master Card/Visa (EMV). 3.

2. What are Smart Cards and Their Benefits? Smart cards a card similar by size to today’s plastic card, that has a chip embedded on it. By adding a chip to the card it becomes a smart card with power to serve many different uses. As an access control device smart cards make personal and business data available only to appropriate users. Another application provides users with the ability to make a purchase or exchange value. Smart cards provide data portability, security and convenience.

There are 2 kinds of smart cards: “ intelligent” smart cards contains a central processing unit-A CPU, that actually has the ability to store and secure information and ” make decision” as required by the card issuer’s specific applications needs. Because “ intelligent” cards offer a “ read / write” capability, new information can be added and processed. The second type is called memory card. Memory cards are primarily information storage cards, that contain stored value, which the user can spend” in payphone, retail, vending or related transactions. The intelligence of the integrated circuit chip

in both types of cards allows them to protect the information being stored from damage or theft. For these reason smart cards are much more secure then magnetic stripe cards, which carry the information in the outside of the card and can be easily copied.

There are also contacts less smart card. Contact less smart card don’t require contact smart card reading, but are recognized by contact less smart card terminal which has to be near by! The technology is already in place to allow the consumers to combine services, such as their credit cards, long distance services, and ATM cards into one card. Smart cards can also perform other functions including providing security for Internet users and allowing travellers to check into hotels. Smart cards provide data portability, security and convenience. Smart cards help businesses evolve and expand their products and services in a changing global market place. Banks, telecommunications companies, software and hardware companies and airlines all have the opportunity to tailor their card products and services to better differentiate their offerings and brands.

The combination of applications available on smart cards also may help them to develop closer relationship with their customers. Some of the benefits include: The ability to manage and control expenditures more effectively, fraud reduction, reduced paperwork and elimination of the need to complete redundant, time consuming forms. The potential of having one card with ability to access multiple services, networks and Internet. The benefits of smart cards are endless and they are our future! 3.

3. Movement for the Use of Smart Cards in a Linux Environment. MUSCLE – Movement for the Use of Smart Cards in a Linux Environment. MUSCLE is a project to coordinate the development of smart cards and applications under Linux.

The purpose of this project is to develop a set of compliant drivers, API’s, and a resource manager for various smart cards and readers for the GNU environment. Many years before, magnetic tape stripe card was used for the necessity for every day life. The magnetic stripe works by encoding identification on a magnetic tape similar to how a computer writes information onto a floppy disk. This method is powerful but insure in many instances. These magnetic cards are easy to reproduce and there is no form of encryption.

The Smart Card is different. By using multi directional interaction between the card and the reader true authentication of identity can be met in a more powerful way that is much more secure than the traditional magnetic stripe card. The cardholders’ identity is held on the card via a secret key. Smart card has a small yet powerful computer built into it. This computer allows the card to interact with the card reader.

Some cards such as the Schlumberger Cryptoflex perform cryptographic functions such as key and certificate verification, encryption, random number generation, etc. Others such as the Schlumberger Multiflex cards perform more general functionality. One of the Multiflex cards, the Cyberflex runs Java binaries on the card making it easier to write high-level programs for the smart card. Most smart cards have little program storage on the card,

roughly 3 – 8K. The 100 mm. CPU contained on a smart card has roughly the same capability as the 1970’s Radio Shack TRS-80 Model 1 Personal Computer.

Although the TRS-80 did not run Java binaries or have built in cryptographic functionality, it did operate at roughly the same speed as many modern smart cards. 3. 4. Smart Card Technology for 2000 and beyondMicroelectronic solutions for multi-application secure smart cards.

MASSC (Multi-Application Secure Smart Card) focuses on the microelectronics part of target applications, including the development of VLSI semiconductor hardware, embedded software and development methodologies. Running under MEDEA (Microelectronic Development for European Applications), the Eureka program or pre-competitive collaborative research and development, MASSC brings together SGS-Thomson Microelectronics, Bull CP8, De La Rue Card Systems, Dyade and OscarD. The partners’ areas of expertise cover the full spectrum of components and system levels required to meet the emerging needs of the smart card market through the development of highly secure and open platforms. The aim of the project is to develop new platforms that will offer:* Advanced VLSI, sub micron chips for embedding in plastic cards;* High levels of security against a broad range of threats, including fraud, accidental errors and failure through degradation;* Easy programmability and application downloading capability;* New and innovative methodologies for fast development of derivatives with security certification; An open and secure platform:* MASSC will develop an open and secure platform for dynamic downloading of applications to be executed on an embedded virtual machine. Security

kernel and a library of encapsulated functions will be included in the platform to offer built-in security at all levels.

The platform will be compliant with industry standards including ISO, EMV, and Java Card;* The chips will be used highly advanced silicon processing with process technologies of 0. 35 down to 0. 25 microns. The core CPU will be a 32 bit processor based on SGS-THOMSON’s leading edge ST20 RISC architecture;* Highly secure and open on-chip embedded software will be developed and it will manage the chip resources at both functional and physical levels, performing cryptographic operations, interpreting high-level code, managing firewall and inter-application security protection mechanisms and bootstrapping the software downloading process; 3.

5 Prepaid Smart Card TechniquesA prepaid smart card is a kind of technology that can contains stored value, which the person holding it can spend at retailers. A system provider is used to let the retailers to reimburse the actual money after they accept the stored value from card. A system provider receives money in advance from people and stores corresponding value onto their cards. During each of these three kinds of transactions, secured data representing value is exchanged for actual money or for goods and services. Card Types* Memory cards – The chip on this card consists only store and a little hardware that prevent the stores data being used by other unauthorized people unless there are password or pins are input correctly.* Shared-key cards – This card allows communication with any device that have the same keys.

The chips are standard micro controller card chips, with masked-in software for the cryptographic authentication algorithms.* Signature-transporting cards – In this card, it store stores publicly verifiable digital signatures created by the system provider and fills them in like blank checks when spending them.* Signature-creating cards – These chips also contain a micro controller, but in combination with a dedicated co-processor capable of making digital signatures. Instead of spending signatures created by the system provider, they create their own.

The memory card is suitable for closed systems where there is little incentive for fraud by persons or retailers. The card cost is low but low security makes it unsuitable for more general use. The signature-creating card offer little fundamental advantage over less expensive cards and it is far too slow in signing for highway speed road-tolls and even some telephones. Shared-key and signature-transporting which is used widely today have same kinds of micro controller chips, and have the same card cost.

The system cost with shared-keys is higher than with signature transporting. The main reason is that shared-keys require tamper-resistant modules at all points of payment and processing sites, while these modules are not needed with signature- transporting. 3. 6. ConclusionsIn this technology world, smart card becomes one of the important technologies in our daily life.

Smart card is now one of the advance technology created. Smart card is just a card with a small chip that is stick to it. Smart cards provide data portability, security and convenience. Smart card is very unique and it can be used as personal data storage, where all personal information can be

store in the smart card. This technology allows company like banks to create ATM card to user which all information is store in the smart card. Some of the smart card benefits are it has the ability to manage and control expenditures more effectively, fraud reduction, reduced paperwork and elimination of the need to complete redundant, time consuming forms.

Now with Java Card technology, it has the ability to create smart card that can be run only platform. Java Card specifications enable JavaTM technology to run on smart cards and other devices with limited memory. With this two new technology, smart will be more efficient and it will be easily implemented into other applications as well. In the olden days, magnetic tape stripe card was used for storing data.

Its method is powerful but it is easy to duplicate and there is no form of encryption to the data. So everyone may use the data. Now with smart card, the smart card used multi directional interaction between the card and the reader. The data is much more secured and smart card is implemented with encryption algorithms to encrypt the data. Only with password or pins will be able to decrypt the data.

Now smart comes with various type of smart card. One type of the smart card that mentions in the literature review is the prepaid smart card. This kind of smart card stores value into the smart card. Amount of money will be deducted from the value in the smart card. With this kind of technology, we will be able to make payment using a smart card.

We don’t have to make any large amount of money where a smart card can pay for everything. Various smart card technologies can make our life easier

by storing data in it, but without a good security, then smart card is no different than a piece of worthless card. A good encryption has to be implemented into the smart card for security. Without that, other unauthorized people will easily see the data in the smart card. Chapter 4: Access Control4. 1 Overview Study Of Access Control ListsAccess Control List (ACL) filters network traffic by controlling whether routed packets are forwarded or blocked at the router’s interfaces.

The router will examine each packet to determine whether to forward or drop the packet, based on the criteria that has been specified within the access control lists. Access control list’s criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information. It is important to note that sophisticated users can sometimes successfully evade or fool basic access lists because that is no any authentication is required. This is the major drawback that needs to be considered. There are many reasons to configure access lists. For example, we can use access lists to restrict contents of routing updates, or to provide traffic flow control.

But one of the most important reasons to configure access lists is to provide security for the network, that is the reason that needs to be focused on. We should use access lists to provide a basic level of security for accessing our network. If we do not configure access lists on our router, all packets passing through the router could be allowed onto all parts of our network. For example, access lists can allow one host to access a part of our network, and prevent another host from accessing the same area. As refer to the figure below, host A is allowed to access the Human Resources network and host B

is prevented from accessing the Human Resources network. Access lists should be used in “ firewall” routers, which are often positioned between the internal network and an external network, such as the Internet.

We may also use access lists on a router positioned between two parts of the network, to control traffic entering or exiting a specific part of the internal network. In order to provide the security benefits of access lists, we should at a minimum configure access lists on border routers. Routers are situated at the edges of the network. This provides a basic buffer from the outside network, or from a less controlled area of our own network into a more sensitive area of our network. On these routers, we should configure access lists for each network protocol configured on the router interfaces.

We can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface. Access lists must be defined on a per-protocol basis. In other words, we should define access lists for every protocol enabled on an interface if we want to control traffic flow for that protocol. (Note: Some protocols refer to access control lists as “ filters”.)4.

2 Lock-and-Key Access ControlLock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists. When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface’s existing IP access

list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first Telnet to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and are able to reach their destination host. Two examples of when we might use lock-and-key are as follows: 1. When we want a specific remote user (or group of remote users) to be able to access a host within our network, connecting from their remote host(s) via the Internet.

Lock-and-key authenticates the user, and then permits limited access through our firewall router for the individual’s host or subnet, for a finite period of time. 2. When we want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, we could enable access to the remote host only for the desired set of local user’s hosts.

Lock-and-key requires the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts. The following process describes the lock-and-key access operation: 1. A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router. 2.

The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The

user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server. 3. When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.

)4. The user exchanges data through the firewall. 5. The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it.

The configured timeout can either be an idle timeout or an absolute timeout. 4. 3 Mosaic 2. 0 User AuthenticationsMosaic 2.

0 allows access restriction based on several criteria:* Username/password-level access authorization.* Rejection or acceptance of connections based on Internet address of client. There are two levels at which authentication information can be passed to the server. The first one will be “ global access configuration file” and the “ per-directory configuration files”. Global access configuration file is not as common as per-directory configuration files. Per-directory configuration means that users with write access to part of the file system that is being served (the Document Tree) can control access to their files as they wish. They need not have root access on the system or write access to the server’s primary configuration files. Also, the per-directory configuration files are read and parsed by the server on each access, allowing run-time re-configuration. The global configuration files are only

parsed on start-up or restart, which usually requires root authority. There is a speed penalty associated with using the per-directory configuration files, but that’s the trade-off you have to take. Access control for a given directory is controlled by a specific file in the directory with a filename as specified by the “ AccessFileName” directive. The default filename is “ htaccess”. In Basic Authentication (lower authentication method) operation, the password is passed over the network, it is not encrypted but not as plain text, it is “ uuencoded.” Anyone watching packet traffic on the network will not see the password in the clear, but the password will be easily decoded by anyone who happens to catch the right network packet. So basically this method of authentication is roughly as safe as telnet-style username and password security — if we trust our machine to be on the Internet, open to attempts to telnet in by anyone who wants to try, then we have no reason not to trust this method also. In MD5 Message Digest Authentication (higher authentication method), the password is not passed over the network at all. Instead, a series of numbers is generated based on the password and other information about the request, and these numbers are then hashed using MD5. The resulting “ digest” is then sent over the network, and it is combined with other items on the server to test against the saved digest on the server. This method is more secure over the network, but it has a penalty. The comparison digest on the server must be stored in a fashion that it is retrievable. It stores the password using the one way “ crypt() function”. When the password comes across, the server uudecoded it and then crypts it to check against the stored value. There is no way to get the password from the crypted value. In MD5, we need the information that is stored, so we can’t use a one way hashing function to store it. This means

that MD5 requires more rigorous security on the server machine. It is possible, but non-trivial, to implement this type of security under the network security model. 4. 4 Configuration Of Access Control ListsEach protocol has its own set of specific tasks and rules required for us to provide traffic filtering, in general most protocols require at least two basic steps to be accomplished. The first step is to create an access list definition, and the second step is to apply the access list to an interface. For the first step, we create access lists for each protocol that we wish to filter, per router interface. For some protocols, we create one access list to filter inbound traffic, and one access list to filter outbound traffic. To create an access list, we need to specify the protocol to filter, we assign a unique name or number to the access list, and we define packet-filtering criteria. A single access list can have multiple filtering criteria statements. When configuring access lists on a router, we must identify each access list uniquely within a protocol, by assigning either a name or a number to the protocol’s access list. When creating an access list, we define criteria that are applied to each packet that is processed by the router; the router decides whether to forward or block each packet based on whether or not the packet matches the criteria. Typical criteria we define in access lists are packet source addresses, packet destination addresses, or upper-layer protocol of the packet. However, each protocol has its own specific set of criteria that can be defined. For a single access list, we can define multiple criteria in multiple, separate access list statements. Each of these statements should reference the same identifying name or number, to tie the statements to the same access list. We can have as many criteria statements as we want, limited only by the available memory. Of course, the more statements we have, the more difficult it will

be to comprehend and manage our access lists. For the second step, in some protocols, we can apply up to two access lists to an interface: one inbound access list and one outbound access list. With other protocols, we apply only one access list, which checks both inbound, and outbound packets. If the access list is inbound, when the router receives a packet, the Cisco IOS software checks the access list’s criteria statements for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet. If the access list is outbound, after receiving and routing a packet to the outbound interface, the software checks the access list’s criteria statements for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet. 4. 5 Log FileMost of the client-server applications are using the log file mechanism to keep track on what the user has accessed. Log file format can be classified into two categories, namely, common log file format and extended log file format. Even though the majority of the analysis tools support the common log file format, but the information about each server transaction is fixed. In many cases, it is desirable to record more information. Sites sensitive to personal data issues may wish to omit the recording of certain data. In addition, ambiguities may arise in analyzing the log file format since field separator characters may in some cases occur within fields. Thus, the extended log file format is designed to meet the following needs:* Permit control over the data recorded.* Support needs of proxies, clients and servers in a common format* Provide robust handling of character escaping issues* Allow exchange of demographic data.* Allow summary data to be expressed. As describe above, the log file format permits customized log files to be

recorded in a format readable by the generic analysis tools. A header specifying the data types recorded is written out at the start of each log. Log file generators should follow the line termination convention for the platform on which they are executed. Each line may contain either a directive or an entry. Entries consist of a sequence of fields relating to a single HTTP transaction. Fields are separated by white space; the use of tab characters for this purpose is encouraged. If a field is unused in a particular entry dash "-" marks the omitted field. The meanings of the fields are defined by a proceeding #Fields directive. If a field is omitted for a particular entry a single dash "-" is substituted. Directives record information about the logging process itself. For example:" Fields: [...]" – Specifies the fields recorded in the log.{Note: The directives Version and Fields are required and should precede all entries in the log. The Fields directive specifies the data recorded in the fields of each entry.}Analysis tools may generate log summaries. A log summary entry begins with a count specifying the number of times a particular even occurred. For example a site may be interested in a count of the number of requests for a particular URI with a given referer: field but not be interested in recording information about individual requests such as the IP address. Log file parsers should be tolerant of errors. If an entry contains corrupt data or is terminated unexpectedly, the parser should resynchronize using the end of line marker and continue to parse the following entries. Entries must not contain any ASCII control characters. 4. 6 ConclusionsIn the network environment, the use of access control is recommended because it may offer a lot of advantages in terms of security, protection and etc. There are several benefits of access controls. First, it can record the details of the user access to the networks. The information is

essential in recording the user usage of the computer. Second, when a certain computer has only one authorized user, the package can use the passwords and encryption methods to ensure that only the authorized user has access to the data and programs held on the machine. However, if more than one person is using the machine, the access control package segregates programs and data so that a user may only see and use the files that the package wishes him or her to view. The implementation of access control in the network environment is very important so that the goals of protection could be achieved. The goals of protection are:* Prevent mischievous, intentional violation of an access restriction by a user.* Ensure that each program component active in a system uses system resources consistently with the stated policies for the uses of these resources. Enforcement of the policies governing resources use. There are varieties of ways involved when policy is concerned. These depend on the hardware being setup and the operating system handled them. Lock-and-key provides the same benefits as access control lists. However, lock-and-key also has the following security benefits over access control lists.* Lock-and-key uses a challenge mechanism to authenticate individual users.* Lock-and-key provides simpler management in large Internet works.* In many cases, lock-and-key reduces the amount of router processing required for access lists.* Lock-and-key reduces the opportunity for network break-ins by network hackers. With lock-and-key, we can specify which users are permitted access to which source/destination hosts. These users must pass a user authentication process before they are permitted access to their designated host(s). Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions. However, it also

has some risks when the lock-and-key method is used. When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user’s address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem. Access control lists are normally searched sequentially to find a matching rule, and ACLs are ordered specifically to take this factor into account. Because of the increasing needs and requirements for security filtering and packet classification, ACLs can expand to the point that searching the ACL adds a significant amount of time and memory when packets are being forwarded. Moreover, the time taken by the router to search the list is not always consistent, adding a variable latency to the packet forwarding. A high CPU load is necessary for searching an ACL with several entries. Chapter 5: Proposed Programming Tools and Techniques5. 1 Background Study of Java Programming ToolsJava was created by a team of programmers at Sun Microsystems in 1991. The main challenges of their job were to create a computer language that could be used to build programs that would run in fundamentally different execution platform ands environments. Because these machines provide a wide variety of hardware and software environments, it was necessary that Java be Platform-independent. This would allow the same software to execute without change on a heterogeneous set of devices. However, java did not become a success. Instead, it was a explosive growth of the World Wide Web and the Internet that caused the Java to receive so much attention. What the

Internet has in the common with consumer electronic devices is its diversity of hardware. The Internet allows many different types of computers to be connected together, including the computers that use fundamentally difference CPUs and operating system. Therefore, the ability to write a portable program is as beneficial to the Internet as it is to consumer electronic devices. There are two types of programs that can be built in Java, which are applications and applets. The feature that is best known about Java is java applet; it can be used to create programs that execute from World Wide Web pages. Java programs made such a big splash on the Web because they offered interactivity in a medium that was largely one way. The Web distributes almost all information in a passive manner. Someone using a browser asks for a page, looks it over, asks for another, looks it over, and so on. A Java applet running on a Web page provides a much richer experience in terms of information and user interaction. Information can change in response to user input or be updated dynamically as a Web page is viewed. Although Web-based programs are strength of the language, Java is a general-purpose language that can be used to develop all kinds of programs, it include designing application for any computer. Besides that, a Java program is also created as a text file with the file extension . java. It is compiled into one or more files of bytecodes with the extension. class. Bytecodes are sets of instructions similar to the machine code instructions created when a computer program is compiled. The difference is that machine code must run on the computer system it was compiled for, and bytecodes can run on any computer system equipped to handle Java programs5. 2 Java ToolsIn order to write Java applications or applets, it needed more than a language so that it could let the programmer to write,

test, and debug the programs. Below are some examples of Java Tools5. 2. 1 CompilerThere is a Java compiler called as javac. The Java compiler takes input source code files and converts them into compiled byte code files. These files typically have the extension. java5. 2. 2 InterpreterThe Java interpreter, known eponymously as java, can be used to execute Java applications. The interpreter translates byte codes directly into program actions. 5. 2. 3 DebuggerThe Java debugger, jdb, enables the programmer to debug the Java classes. Unfortunately, the Java debugger is a throwback to the pre-GUI debugger dark ages of programming. However, it can use the jdb to set breakpoints, inspect objects and variables, and monitor threads. 5. 2. 4 DisassemblerThe Java Developer’s Kit includes a disassembler, javap that can be used to display the public interface, both methods and variables, of a class. Additionally, the Java disassembler includes options to display private members or to display the actual byte codes for the class’s methods. This last option can be particularly useful if the programmer want to achieve a greater understanding of the byte codes used by the Java interpreter. 5. 2. 5 Header File GeneratorBecause Java is a new language and must fit in a world dominated by C and C++, included in Java is the capability to use native C code within a Java class. One of the steps in doing this is using the Java header file generator, javah. 5. 2. 6 JavadocThe programmers fought that it in every way possible. Unfortunately, there is no longer any excuse for not documenting the source code. Using the JavaDoc utility provided with the Java Developers Kit, programmers can easily generate documentation in the form of HTML files. To do this, programmers embed special comments and tags in the source code and then process their code throu