# The CHOICE Network: Broadband Wireless Internet Access In Public Places

Victor Bahl
Anand Balachandran
Srinivasan Venkatachary

February 2000

Technical Report
MSR-TR-2000-21

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

# Acknowledgements

We are thankful to our colleagues who helped crystallize our initial ideas and designs. In particular we are grateful to Pierre De Vries from Microsoft Advanced Product Development Group and to Stephen Dahl from Microsoft Research for their help in getting this system deployed.

# Broadband Wireless Internet Access In Public Places

**Paramvir Bahl**
Microsoft Research
bahl@microsoft.com

**Anand Balachandran**
University of California San Diego
abalacha@cs.ucsd.edu

**Srinivasan Venkatachary**
Microsoft Research
cheenu@microsoft.com

## Abstract

***We have built a network, called the CHOICE network, which globally authenticates users and then securely connects them to the Internet via a high-speed local area wireless network. Our network provides easy-to-use, individual-centric, service-oriented wireless Internet access in places other than the traditional corporate offices and homes. Our architecture is hardware and protocol agnostic and is built on an easily deployable software module called the Protocol for Authorization and Negotiation of Services or PANS. PANS provides authorization, access, privacy, security, policy enforcement, quality of service (QoS) and accounting. In this paper, we describe PANS in detail. We discuss our design decisions, system operation, implementation and performance. We evaluate PANS and show that it is scalable and secure. Our network has been deployed and is operational at a local mall in Bellevue, Washington.***

## 1 Introduction

In an increasingly fast-paced mobile society we invariably find ourselves spending a considerable amount of time in public places such as airports, hotels, libraries, and shopping malls, and at public events such as conferences, meetings, lectures etc. Yet, Internet access is not typically provided in these settings. When it is provided it is generally in the form of hardwired fixed kiosks that are not very convenient, or via wide-area wireless data networks that are excruciatingly slow. Consequently, there is much enthusiasm around the impending deployment and availability of the so-called “third-generation” or 3G wide-area cellular networks. These networks, which will be available within 1 to 3 years, are touted as the wave of the future because of their ability to support data networking at speeds of up to 2 Mb/sec per cell for a non-roaming user [1], where a typical cell is between 1 to 2 miles in radius. This means that the maximum achievable *raw data rate* that an individual can get with 3G networks is around 2 Mb/sec when he or she is the **only** user of the network in a 1-2 mile radius! This is not realistic. A more compelling scenario is one where there are many users in the cell, in which case the average throughput that any one individual gets may be substantially below the 2 Mb/sec advertised speed.

Today, wireless local area networks (LANs) can provide data connectivity at up to 11 Mb/sec per access point [2] [3], within 1 to 3 years they will provide access speeds of up to 54 Mb/sec [4] and looking beyond 3 years this data rate is expected to reach 100 Mb/sec. Consequently, there is and will continue to be a *25X* difference in data transmission speeds between wide-area and local area wireless networks. It is our thesis that as users become accustomed to the higher speed wireless LANs and as the Internet becomes increasingly multimedia-centric, the frustration level with the much slower 3G system will increase.

Consider now how people traditionally connect to the Internet. An individual may have a personal account with an Internet Service Provider (ISP) in which case she uses her home computer to establish a link with the ISP through a modem or special communication line. In this situation her access is tied to her wired link provider, or to the ISP through which she has her account. Alternatively, she may access the Internet at work or elsewhere through a network that is provided and maintained by her employer. In this situation her access is tied to her employer. Neither of these paradigms provides the individual the freedom to access the Internet from any location without any dependence on a particular ISP or her company. We believe that it is desirable to eliminate the dependence of Internet access on either or both of these elements.

We have built and deployed a network, called the *CHOICE* network, which provides a *choice* to individuals in how they access the Internet from "almost anywhere". Almost anywhere includes places of congregation or public spaces such as shopping malls, airports, restaurants, libraries, hotels, train-stations etc. Using widely available standards-based wireless LAN technology [2] the CHOICE network provides Internet access at speeds that are *25X* greater than 3G speeds, to individuals who present the proper identification. Thus, any individual entering a public building, which has a wireless LAN is able to access the Internet painlessly. In addition, we have designed the CHOICE network to offer policy-based services like different levels of privacy, last-hop QoS, and flexible charging. Additional services such as accessing local resources such as printers, locating buddies, and electronic in-building navigation can also be offered as part of this network.

The software language of the CHOICE network is the *Protocol for Authorization and Negotiation of Services*, or PANS. PANS facilitates authentication, authorizes access, enforces policy and last-hop QoS, and provides privacy to network users and accounting to network operators. The user can be anywhere in the world and PANS can securely authenticate her credential using a globally available database. PANS allows Internet access in accordance to a pre-configured policy manager. Last-hop privacy and security is based on per-user dynamically generated varying-length keys, which are valid for a varying amount of time. In this paper, we describe the design, implementation, operation, and performance of PANS. We focus only on the authorization, access, privacy and security aspects of PANS.

The rest of this paper is organized as follows: In Section 2 we describe the problems and difficulties in using current wireless LAN technologies in public spaces. In Section 3, we describe the CHOICE network. We outline the design goals, system components, system operation, and implementation. In Section 4, we evaluate PANS focusing on security and scalability. In Section 5, we look at scalability issues and PANS performance. In Section 6, we survey related work in the field. In Section 7, we describe on-going and future work including on-going deployment of our system in a local mall. Finally, we conclude in Section 8.

## 2 Problems in Deploying Current Wireless LANs in Public Places

For several years now in events such as the Internet Engineering Task Force (IETF) quarterly meetings and the Institute of Electrical and Electronic Engineers (IEEE) meetings, sponsoring companies and event organizers have taken it upon themselves to set up temporary wireless LANs for attendees to use for connecting to the Internet. Unfortunately, expanding this initiative to a wider setting in a public place has not been easy due to the following reasons: First, current solutions require users to somehow determine the operational parameters of the local host network and then to configure their personal computers and wireless network cards to use these parameters[1]. For wide scale adoption this requirement is unreasonable since most users are non-technical. Second, currently it is difficult for network operators to guarantee adequate privacy and security to end-users who want to complete their transactions without fear of eavesdropping and masquerading. Third, network operators have found it difficult to protect their network from malicious users who intend to disrupt the system. Fourth, currently there is no convenient mechanism available to the host organization, which allows them to offer a choice of different service levels to end-users, and fifth, network operators find it difficult to determine how much bandwidth is being used by a particular user and to enforce policies on bandwidth usage on an individual basis. Consequently, the economic incentive for the host organization to deploy a public network that offers multi-level service with creative charging has been less than compelling.

Currently available wireless LANs limit themselves to the problem of user authentication, privacy and security via either (1) MAC level filtering and/or via (2) shared key authentication and privacy, both of which are layer-2 mechanisms. In MAC level filtering the access point (AP) maintains a list of valid MAC addresses. For each incoming packet on the wireless segment, the AP checks the source MAC address against the list of valid addresses in the table. If there is a match, the packet is forwarded to the wired segment otherwise it is dropped.

A number of problems emerge when such a layer-2 filtering mechanism is employed for keeping out unauthorized users in a public setting. First, the system administrator has to somehow know the MAC address of the user's wireless adapter. He then has to enter this address in a central database or alternatively into every AP in the organization as a valid address. Second, the number of authorized users can be potentially very large, much larger than what most APs can currently hold. Third, if the user loses her wireless adapter anyone who finds this adapter can use it to gain access to the network. Fourth, MAC-level filtering does not protect against hardware address spoofing and hence does not provide privacy and security to the users of the network as explained below.

Unlike a wired LAN a RF wireless LAN is difficult to secure. This is because RF signals are not restricted to well-defined boundaries; an RF node can "listen" to transmissions from other RF nodes operating on the same frequency within its transmission range[2]. Furthermore, an unauthorized user can gain access to the network by masquerading as a valid user through address spoofing. This is clearly undesirable; consequently, an encryption mechanism that provides security and privacy is needed. This then motivates the second method that is available today, which depends on using shared keys.

Wireless LAN standards such as the IEEE 802.11 [2] and HomeRF$^{TM}$ [5] include an optional provision for authentication and privacy that is based on shared key authentication. In 802.11 this mechanism is called the *Wired Equivalent Privacy* (WEP) function. A shared key is configured into the AP and its wireless clients ahead of time. In 802.11 products with the WEP option enabled data is encrypted before it is sent wirelessly using a 40-bit encryption algorithm known as RC4[3]. The same key is used for both authentication and encryption / decryption of data; thus only wireless clients with the exact shared key can correctly decipher the data.

Once again problems emerge when such networks are deployed in a public place. First, distributing keys in a public environment is relatively harder than in a corporate setting. Second, keys have to be deployed on a per-user basis to maintain security. Currently there are no simple mechanisms for generating and distributing keys for last-hop privacy. Third, even per-user keys have to be changed frequently since the algorithm can be broken in time [6]. Changing keys frequently is not convenient with current products.

A popular layer-3 mechanism for secure communications on the Internet that can also be used in wireless LANs is called IP Security (IPsec) [10]. Briefly, IPsec includes two mechanisms for secure data transfers on the Internet, viz. IPsec authentication header (AH) [26] and IPsec encapsulating security payload (ESP) [27]. IPsec AH

[1] Configuration parameters could include setting the adapter's SSID, setting its connection mode and operational channel, setting the security option and key, using a specific login sequence, etc.

[2] Even when the physical layer of the network is based on spread spectrum communication, the system is insecure as it is relatively easy for a malicious user to scan all channels and determine the hopping pattern and *Network Identifier* of the target network.

[3] Several companies have recently announced products that use 128-bit keys.

provides authentication of data origin, data integrity, and protection against replay attacks. IPsec ESP provides confidentiality of data via encryption and can optionally provide authentication. Both mechanisms can be configured to work in one of two modes: transport mode or tunnel mode. Transport mode provides end-to-end security while tunnel mode provides security between the two end points of the tunnel, which may not be the same as the end-points of the connection. IPsec AH and IPsec ESP combined suitably with a secure key exchange mechanism like IKE [28] can authenticate users identified by a certain fixed IP address and guarantee confidentiality of data transferred.

However, a system based on IPsec did not fully satisfy all of our requirements for specific reasons we outline below. First, we found that IPsec is not as widely available as we would like it to be. For example, IPsec is not available in Windows 95, Windows 98, Windows CE, and several versions of MAC OS. Second, IPsec couples user keys and security association very tightly with IP level information. This directly impacts our goal of supporting roaming users whose IP address changes frequently. In the CHOICE network, as we will describe in subsequent sections, we identify users by a specific (key, token) pair that is completely de-coupled from IP level information and consequently, support for mobility with fast hand-offs is an integral part of the network. Third, we found that most IPsec implementations that are available today exchange keys based on pre-configured machine certificates whereby the machine endpoint is authenticated and not the user. In the event that multiple users use one machine, authorizing access based purely on machine certificates creates a security loophole [29]. Fourth, we explored the possibility of IPsec tunnel mode between the mobile client and AP by pushing IPsec functionality into the AP and treating it as a security gateway. However we did not find a single vendor who currently supports IPsec functionality in the AP. Finally fifth, we wanted a mechanism that was protocol agnostic, so we could support both IP and WAP [30] devices at the same time. IPsec is tightly linked to the IP protocol.

In summary, while current layer-2 wireless LAN technologies do not provide adequate levels of security and privacy, layer-3 technologies like IPsec are not widely available, do not support mobility and are not protocol agnostic. We were motivated by a desire to empower the mobile individual by providing her with choices when she accesses the Internet wirelessly in a public place. To do this expeditiously, we concluded that a lightweight mechanism that provides authorization, access control, privacy, security, local mobility, accounting, and last-hop QoS was needed. Furthermore, we wanted a mechanism to be both protocol and hardware agnostic so we could support both IP and WAP devices and to run over legacy wireless hardware if needed. The CHOICE network that we describe below contains such a mechanism.

# 3 The CHOICE Network

We now describe the architecture, design points, system components, system implementation, and system operation of the CHOICE network.

## 3.1 Design Considerations

In this section we outline the goals that influenced the design of the CHOICE network. In a subsequent section we describe the details of the various network components that were developed to meet these design goals.

### 3.1.1 Ease-of-Use

Ease-of-use was a dominant design goal for us. We wanted the users to be able to plug their wireless adapters into their personal computers and with minimal effort become connected to the Internet. We thought through usage scenarios and ironed out details on how the PANS software would be installed on the user's machine and opted for a web-based interface for all user interactions with CHOICE. For login we decided to let the host organization's web server redirect the user's browser to an authentication database. This procedure is similar to the one adopted by the designers of the SPINACH system [22].

### 3.1.2 Individual-centric Access

Our goal in designing CHOICE has been to create a system that can establish the identity of hitherto unknown users and allow them access to the Internet and to local resources[4]. Using a trusted database that is globally available solves the problem of establishing the identity of the individual electronically. In addition, it obviates the need for the individual to have any dependency on the ISP that they have signed an agreement with or the company that they work for, when in a public place. This is one of the key ideas in the CHOICE network. The global authentication database can be operated on behalf of many organizations or businesses that might want to authenticate users. MS-Passport [11] is an example of such a service that is available to users worldwide.

It should be noted that our choice of using a global database does not preclude using a locally available database instead. In fact when CHOICE is deployed in the enterprise, the authenticating server is on the local Intranet and the user's identity is established by a locally available corporate database that confirms that the individual is an employee of the corporation.

### 3.1.3 Hardware and Protocol Independence

Another important design goal for us was to keep our system hardware and protocol agnostic. This is in contrast to some other approaches, which are tied explicitly to the underlying media [21] or to higher layer protocols (e.g. IPsec). Even though we decided to build our system over a standards-based wireless LAN our design does not preclude deployment over other technologies including wireless WANs and wired LANs.

We wanted PANS to be independent of the higher layer protocols. With some technology forecasters predicting that in the next 4 to 5 years there will exist billions of phones

[4] Giving the network operators the ability to establish the identity of the individual who is using their network provides them with some recourse against malicious users.

and PDAs using WAP, it became important to us that our network had to work well with both IP and WAP devices.

### 3.1.4 Privacy and Security

In CHOICE we focused on last-hop security only. The goals that dictated our design were as follows: First, only authorized user should be allowed access to the network and its resources. Second, no one except the user and the authentication database should be privy to personal information such as username, password, credit-card information etc. Even PANS should not be privy to the exchange of information during the authentication phase. Third, the network should provide several-levels of security. An authorized user should be able to choose varying levels of security either actively in real-time or via a pre-configured policy. The basic mode of security (provided by default) would encrypt only the minimal amount that is absolutely necessary for secure operation of the network. However, additional modes such as medium mode (header encryption) and full mode (full packet encryption) should be available as advanced security options, generally fulfilled at an additional expense to the user. Fourth, we wanted dynamic generation of per user keys, which are valid for a varying amount of time and finally, fifth, we did not want to be locked into using any one single encryption algorithm.

### 3.1.5 Policy Management and Service Negotiation

We envision host organizations wanting to exert a finer grain control over how their network is used. In most current wireless LAN deployments known to us all users are treated equal. There is no provision for users who are willing to pay extra to get better network service from the host organization We wanted to build hooks into PANS so that it could be configured according to the policies set by the host organization regarding the level of service, security, and resources to be granted to a particular user. We wanted to allow enforcement of policies that might have been pre-negotiated between the host organization and other corporations, or just general policies regarding different classes of paying and non-paying users.

A policy manager would contain one or more policy tables that define various access and service policies for various classes of users. For example, a corporation might negotiate a service package for free network access at a local airport, for all its employees. The Policy Manager then maintains an entry in its policy table that indicates that employees of this corporation are to be granted access to the Internet via the airport's wireless LAN at the negotiated level.

### 3.1.6 Accounting and Quality of Service Management

Since security forced us to think about per packet processing, it was easy for us to incorporate per packet accounting for each user. We decided to let PANS keep track of the number of packets it had processed on behalf of each user. Accounting for packet has some advantages: (1) by keeping track of the number of data packets/bytes sent by each user, PANS allows the host organization to create flexible charging plans and bill users accurately for the amount of bandwidth they have used. Additionally, it is also useful from the standpoint of assessing the collective system demand of members of various organizations that have negotiated service level packages for their members. (2) Accounting information is used by PANS to schedule packet transmissions from the clients in a manner that ensures that QoS guarantees are adhered to as explained below.

Once again, because we had decided to go with per packet processing, the QoS mechanism serendipitously fell out of this design. In its simplest form we envisioned that as part of the key distribution phase the client is notified about the data rate it can operate at. This rate would depend on the policy governing the user or on the real-time negotiation between the user and the host organization. The client and the server side modules will then ensure that this data rate is adhered to. Misuse of the network will be prevented because of the accounting and filtering mechanisms built into the system. A significant decision that simplified our design considerably was to provide per-user QoS as opposed to per-flow QoS (where there can be multiple flows from the same user). In this model user actions prioritize user transactions [5].

## 3.2 System Components

The CHOICE network has several components that manage address allocation, authentication, authorization, security, accounting, and last-hop QoS. These components are illustrated in **Figure 1**.

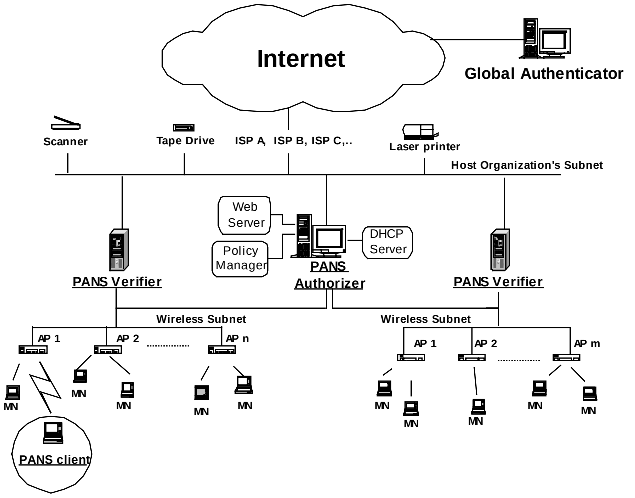


**Figure 1**: The CHOICE network architecture (note: ideally the *PANS Verifier* should be resident at every AP)

The DHCP server [7] leases out an IP address to the client wishing to connect to the Internet. The client is authenticated by the global authentication database, procures a key from the PANS authorizer, and is given access to the network. The key is the signature that is expected in every packet that comes from the user into the CHOICE network. We describe each component in detail below.

[5] In this paper, we do not discuss the implementation of the QoS module. Instead, we simply describe how QoS influenced our design and how it is possible in our system. We focus on the authentication, access, privacy, and security aspects of the network.

### 3.2.1 MS Passport

We use MS Passport [11] as our authentication database. Several factors motivated our choice of Passport as the authentication service. First, its wide availability enables us to offer our service to several millions of users. Second, all transactions with Passport are web-based thereby greatly enhancing the usability of the system for the layperson. Third, and most importantly, all these transactions are carried out over *https*, which is encrypted using the Secure Socket layer (SSL) [12]. This means that there is an end-to-end secure channel between the user and the authentication service. Even if PANS were to be set up by an un-trusted third party, this party cannot decrypt the user's name and password while it is being supplied to Passport.

### 3.2.2 Address Allocation and Naming

The CHOICE network uses a standard DHCP server to lease out IP addresses to potential clients on the public sub-network. The IP address scope and the lease period are configured by the host organization at setup time. One drawback of DHCP is the limited scope within which the server can lease IP addresses. This problem can be overcome by using a Network Address Translator (*NAT*) [8] in preference to DHCP. A NAT server allocates a non-routable private network addresses to clients and then translates these addresses to a public IP address for communications with the Internet.

The Web server is the user's entry point into the CHOICE network. It is through the web interface that the user begins the authentication process. The CHOICE network Web server is based on Active Server Pages (*ASP*) [9] and guides the user through the authentication process. It would be evident that a prerequisite to the authentication process is the successful obtainment of a valid IP address and a connection to the Web server. Since both these network connections have to go through prior to authentication, the task of the PANS server module is divided between two sub modules, which we discuss in the subsections below.

### 3.2.3 PANS Authorizer

The first sub-module, called the *PANS Authorizer* authorizes the client's access to the network, upon successful completion of authentication. In addition, it handles the task of determining service policies, generating keys, and communicating service levels and keys to the clients and to the *PANS Verifiers* (to be discussed next). The PANS Authorizer performs IP-level filtering based on the destination IP-address of each packet. Any packet with a destination address other than the DHCP server, the DNS server, the Web server or the Passport server is dropped.

Upon authentication (by MS Passport), the PANS Authorizer does a look-up in its Policy Table to determine the users service level, generates a (*key, token*) pair, and then communicates the service level $L_n$ and the (*key, token*) to the *PANS Client* module residing on the users mobile host and to the PANS Verifier. In addition, the client and the verifier also get a *key_id*, which is an index into an array of valid (key, token) pairs that have been given out to the clients.

Each (key, token) pair is valid for a finite amount of time after which the user must renew her identity and obtain a new pair. The key is the value used for encryption/decryption and the token is the value that is tagged to every packet before encrypting it with the key. Once the user has been authenticated, all her communication is directed through the PANS Verifier, which has knowledge of valid (key, token) pairs that it can be provided with, with every incoming packet from the mobile hosts.

### 3.2.4 PANS Verifier

The second sub-module on the server side, called the PANS Verifier, handles the tasks related to per-packet verification, accounting and policy enforcement on packet transmissions between the mobile users and the public network. We have chosen to separate the tasks of authorizer and the verifier (see Figure 1) in order to achieve a clean separation in the time scales of their operation. The tasks performed by the authorizer are as frequent as the number of new authentication operations, either due to new users in the network or due to an old user having timed out. On the other hand, the PANS Verifier is actively processing each packet that is sent out of the mobile host and runs on a much smaller time scale as compared to the Authorizer. The task of the PANS Verifier includes checking if each packet from a client (identified by a unique key_id) contains the right (*key, token*) combination that the PANS Verifier has in its table entries. In addition, the Verifier keeps an account of the number of packets per user it has serviced and enforces policies such as QoS service-level by dropping packets from a user who violates her service agreement. The separation between authorizer and the verifier was additionally motivated by the need to support a greater number of PANS Verifiers as we introduced support for roaming. This would mean replication of the PANS Verifier; one for each subnet of wireless access points.

### 3.2.5 PANS Client

The final module of the CHOICE network is the one that sits on the user's mobile host. Since the user has access to the PANS Authorizer service prior to authentication, she can download and easily install the PANS Client module in her machine. After this is done, all packets going out of the client either use a default (key, token) and a key_id combination (before authentication) or the combination provided by the PANS Authorizer after successful authentication. In our implementation section, we describe the communication between the PANS client and verifier modules in detail and also give an overview of the PANS protocol.

## 3.3 Implementation and Operation

Having described the components that make up the CHOICE Network in detail, we now describe the detailed implementation of the PANS protocol, and the communication between the PANS client and server modules, using a typical system usage scenario.

### 3.3.1 The PANS Intermediate Driver and User-level Module

PANS is implemented as an Intermediate Miniport driver within the Windows Network Driver Interface Specification (NDIS) protocol stack [13].

**Figure 2** depicts the location of the PANS intermediate driver relative to the NDIS protocol stack. The modular design of NDIS allows us to write PANS as an NDIS Intermediate driver that plugs in to the stack seamlessly. Intermediate drivers are typically layered above the NIC driver and below the transport driver. Thus, we could program PANS to manipulate packets that are delivered by NDIS from the protocol driver above, down to the NIC.

In addition to the kernel-level module, PANS has a user-level module as well. The user level module handles the key and service-level exchanges with the PANS Authorizer after authentication. We have implemented the transfer of keys and service-levels using ASP scripts running on the Web server [9]. Once it has received the key, the user-level module communicates this key (along with the token and `key_id`) to the intermediate PANS driver in the kernel using a simple *ioctl* call. The phenomenon of key exchange with the PANS Authorizer is identical for both the PANS Client and the PANS Verifiers. While the client module uses the information for encryption, the verifier module uses the same key for decryption once the packet is received at the PANS Verifier.

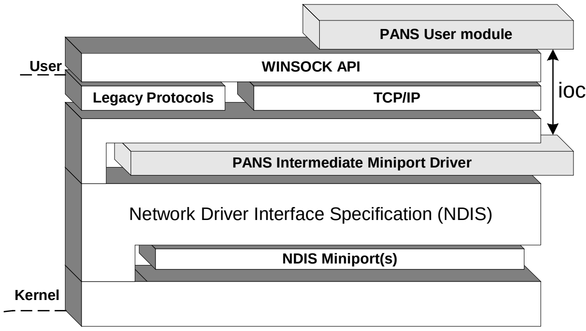


**Figure 2**: The network protocol stack showing the PANS Intermediate driver and its location in the NDIS stack. The PANS user module runs at the application level and communicates with the driver using *ioctl* calls.

The PANS Authorizer contains a daemon that periodically broadcasts a *PANS Authorizer Available* message. This message is to let the PANS client know when it has moved out of the CHOICE network. When the client does not get this message it issues an *ioctl* to the PANS module to stop tagging the outgoing packets and to clean up the routing tables. This is necessary if the client is to operate correctly outside the CHOICE network.

### 3.3.2 The PANS Protocol

PANS is deployed as a software module in the client, the authorizer, and the verifiers. The user-level module handles the web-based exchange of keys and tokens, which in turn are handed down to the PANS driver.

Each successfully authenticated user is given a (`key, token`) pair and a `key_id`: a combination that is used in all future network transactions. This unique combination forms part of the *tag* that is appended to every packet that goes out from the client's mobile host. The typical structure of the PANS protocol tag is shown in Figure 3. As an outbound packet on the client is handed down the protocol stack from the transport layer above, the PANS driver tags it with a *PANS_TAG*. The PANS_TAG is composed of two parts, an unencrypted part and an encrypted part. The first unencrypted part (*Cleartext*) contains the version number, the `key_id`, and encryption type. The encrypted part contains the token (given by the PANS Authorizer) and an MD5 checksum of the data and the PANS_TAG. The checksum field is filled with zeros prior to its computation. The purpose of the MD5 checksum is to ensure authenticity of the data origin and to prevent the packet from being modified in transit (see Section 4.6). This information is encrypted using the secret key that is part of the (`key, token`) pair obtained from the authorizer. We use the triple-DES (Data Encryption Standard) algorithm [14] for encryption. However, our implementation is modular enough to accommodate any other encryption algorithm as indicated in the encryption type field[6].

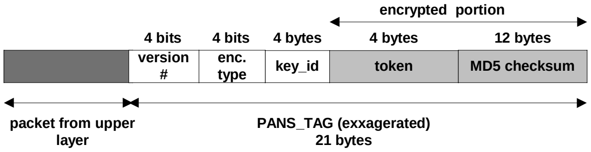


**Figure 3**: The PANS_TAG showing the different fields. The version number, encryption type and `key_id` form the unencrypted portion, while the token and sequence number are encrypted using the encryption algorithm specified under the encryption type.

We made several design decisions in developing PANS. The transport and network layers fragment a packet handed down by the application based on the MTU size of the network. Since the PANS driver adds the PANS_TAG to the packet, we could potentially overshoot the network MTU. In order to fix this problem, we added code in our intermediate driver to report a reduced MTU value (reduced by the size of the PANS_TAG) to the upper layer.

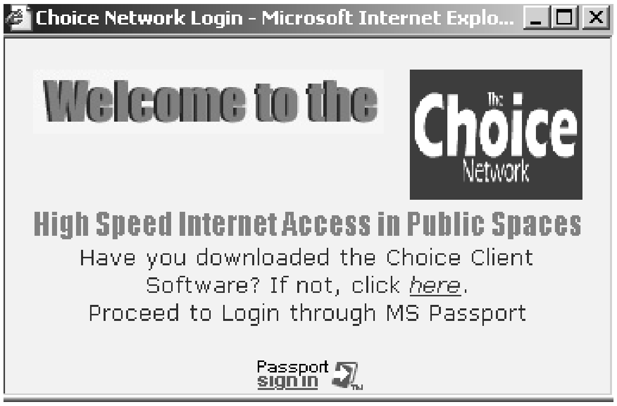
We *tag* the protocol information to the tail end of the packet. This is because the transport layer hands down the packet with a pre-allocated 14-byte header, which is a placeholder for the MAC-level information. The network interface driver fills in the values in this memory area just before the packet is sent out on the link. Adding the PANS_TAG to the beginning of the packet would disrupt all the MAC-level information that is to be interpreted at the receiver end resulting in the packet being dropped. The only way to avoid this problem would be to splice the packet as it is received from the protocol driver above and insert the PANS_TAG at an offset of 14 bytes. Since this involves an intensive data copying operation, we did not go with this implementation.

### 3.3.3 Usage Scenario

[6] For example, the ECC encryption algorithm is much less CPU resource intensive than DES [15]. So this may be a better choice for battery constrained wireless devices. We can accommodate both DES and ECC in our design.

The following is a typical usage scenario of the CHOICE network. The user walks into a public place where the CHOICE network is operational and boots up her mobile host. The DHCP server in the network picks up her DHCP broadcast request and issues a routable IP address. The user now launches her web browser and points it to the default name "*Choice*", which is resolved by the local DNS server to the address of the CHOICE network web server. At this stage the PANS client component can be downloaded and installed on the mobile host from the Web server. Upon installation, the PANS client module is automatically triggered to tag every outgoing packet with a PANS_TAG. However, until the user has authenticated herself to the PANS Authorizer, she does not have a valid token and a valid encryption key to use in the PANS_TAG that is tagged to packets sent out from her host. Nevertheless, a (`key, token`) pair and a `key_id` are required even for the authentication packets. Therefore, in our implementation we have a default (`key, token`) pair and a default `key_id` that the mobile hosts can use up until PANS authenticates them.

The user's web browser is redirected to the CHOICE network logon page as shown in **Figure 4** wherein she types her identity and sends it to MS Passport over `https.` After authentication, the user's web browser now refreshes to a page from where she can obtain her access key and token by means of a single click. The PANS Authorizer gives out the access key and token to the PANS Verifiers as well. All further communication is channeled through the PANS Verifier and is monitored on a per-packet basis.



High Speed Internet Access in Public Spaces

Have you downloaded the Choice Client Software? If not, click *here*.

Proceed to Login through MS Passport

**Figure 4**: The first screen the user sees after she types in "*http://Choice*" on her web browser.

# 4 System Security Evaluation

In this section we discuss how PANS deals with some of the common known threats on computer networks. We first describe the threat and then discuss how we guard against it.

## 4.1 Replay Attack

In this attack, the user's login sequence is monitored (possibly with a protocol analyzer) and recorded by an intruder for playback at a later time. The intruder wants to use this to fool the login server into authenticating a valid user without the intruder necessarily having to know the user's name or password. The intruder sends the login server the same *encrypted or hashed name* or password that the server had accepted in the past.

In CHOICE, username and password are sent to the global authenticator (MS Passport) using `https`, which uses `http` and SSL. Replay attacks are avoided in SSL by varying the encryption or hashing process so that each session is unique and can't be duplicated or repeated.

## 4.2 Masquerade

An unauthorized user pretends to be a valid user. One way to do this is by IP and hardware address spoofing. The IP address and/or the MAC address of a trusted system is assumed and used to gain access rights. For example, an intruder could use address of an authorized user, or the intruder could pretend to be the PANS Authorizer.

Address spoofing (both IP level and MAC level) where the intruder masquerades as an authorized user will not work in CHOICE since the PANS Verifier relies on a token and key pair given to a valid user by the PANS Authorizer at the time of authentication. The authorized client encrypts this token with the key and sends it as part of every transmitted packet to the PANS Verifier. Access is granted only if the PANS Verifier is able to decrypt and match the token properly.

An unauthorized user cannot masquerade as a PANS Authorizer. Each PANS Authorizer has a X.509 certificate [16] issued from any one of several certification authorities [17]. This is then used by the PANS Authorizer to authenticate itself to the PANS client, as a basis for giving it a key. An intruder will not have such a certificate and hence the client will not accept it as a valid PANS Authorizer.

It would be very difficult for the intruder to masquerade as an AP. First he would need to access to the wired network, it would then have to disconnect the AP it was masquerading as and then even if he succeeded, it would not be privy to any user information since that is secure end-to-end.

## 4.3 Repudiation

Much of network-based business and electronic commerce relies on the ability of message or transaction recipients knowing for certain the identity of the sender. This is also important for the purpose of billing by the host organization. Since the CHOICE network is not gullible to masquerading, knowing who originated the message or transactions, or who is using the network is not a issue.

## 4.4 Denial of Service

The CHOICE network is unable to avoid denial of service attacks. However it can detect that a denial of service attack is in progress. The PANS Verifier can keep track of the number of failed attempts to get a packet through and the Authorizer can keep track of the number of IP address requests that have not been subsequently authorized. If there is reason to believe that a denial of service attack is in progress, PANS can raise an alarm and possibly shut down the part of the network where the attack is taking place. In addition, because of the fact that this is a wireless network, the part of the building where the attack is originating from can be identified[7]. A single attacker cannot saturate the backbone network with unauthorized packets since the bandwidth in the wireless segments is at least 10 times slower than the bandwidth in the wired backbone. It would take a coordinated effort between multiple attackers who are geographically separated so they are connected to different APs, to saturate the backbone network, and overload the PANS Verifier, and the PANS Authorizer.

### 4.5 Data Interception and Confidentiality

Last-hop data confidentiality is available in CHOICE via enhanced security modes. The user has the option of encrypting the entire application data before it passes over the wireless network to the PANS Verifier. CHOICE uses different keys for each session ensuring that the encryption format is unique for each network session. The keys carry expiration times, after which the user is asked to download a new key. Discarding keys on a regular basis makes cryptographic attacks more difficult than if the same keys were used with large amounts of data.

### 4.6 Manipulation and Data Integrity

The question here is, can an intruder hijack a packet, change the data and then transmit the packet? One way an intruder can perpetrate such an attack would be to use two directional antennas, one pointed at the client and one pointed at the AP, and bombard the AP with fake packets so that the AP can't receive the packets sent out by the client (since the channel is jammed). The intruder now ends up with the client's data, replaces it with his own and sends it across to the network.

Our defense against this "man-in-the-middle" attack problem is the MD5 checksum that is encrypted and sent as part of the PANS_TAG. Since the MD5 is computed both on the data and the tag, it is unique for every packet. Therefore, the intruder who intercepts the client's data cannot in any way manipulate the packet and use the PANS_TAG to his advantage.

In concluding this section, we believe that the PANS protocol provides adequate protection against all the well-known security threats. Stated another way, PANS is as secure as the underlying cryptographic algorithm. Next, we evaluate PANS scalability.

## 5 PANS Performance

The task of per-packet verification by decrypting a packet and subsequently checking for the client's valid signature in the PANS_TAG, adds value to the CHOICE network and makes it intrusion-proof. However, intercepting a packet, which is in transit through the network could increase the latency incurred in its transmission and also bring down the throughput of the system. In order to verify our hypothesis, we ran a few benchmark tests on the PANS Verifier to measure its throughput, CPU utilization and its effect on the packet round trip time.

### 5.1 Experimental Setup

We use a common simplified setup for all our experiments. This setup is shown in **Figure 5**. Specifically, we use a wired 100 Mb/sec connection between each of the systems. We did this to push more data on to the network and potentially stress the PANS Verifier to its limit. We assigned static IP addresses to the client machines. The system we used to send packets on the network was a 450 MHz Pentium II Dell Inspiron 7500 laptop with 256 MB RAM and a 3COM Megahertz 10/100 LAN Cardbus PC card. The PANS Verifier was a 450 MHz Pentium II Dell Precision 410 workstation with 128 MB RAM and two network interfaces. One interface (a 3COM 3C918 Fast Ethernet Integrated controller) connected to the subnet of the sender machine while the other (a Cabletron DE 500B PCI Fast Ethernet Controller) connected to the receiver subnet. The receiver machine was a 366 MHz Pentium II Toshiba Tecra 8000 laptop with 256 MB RAM and a Xircom Cardbas Ethernet II 10/100 PC card. All systems were running Windows 2000.

The two programs that we used to study the performance of PANS were `ttcp` [18] and `tcpperf`. Ttcp is generally used for characterizing the performance of the network stack by measuring end-to-end throughput. It outputs the total throughput of the connection and the time taken to send all the packets. `tcpperf` is similar to `ttcp` except that the sending of packets is synchronous. It outputs the average round trip time for transmission of a packet between the sender and receiver.

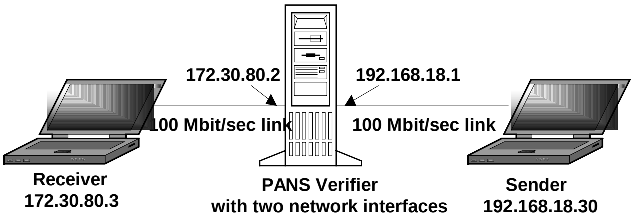


**Figure 5**: Setup showing our private network, which we used to conduct performance studies on the PANS Verifier.

### 5.2 Throughput and CPU Utilization of the PANS Verifier

We measured the throughput of the PANS Verifier using `ntttcp`. The program by default sends 20K buffers

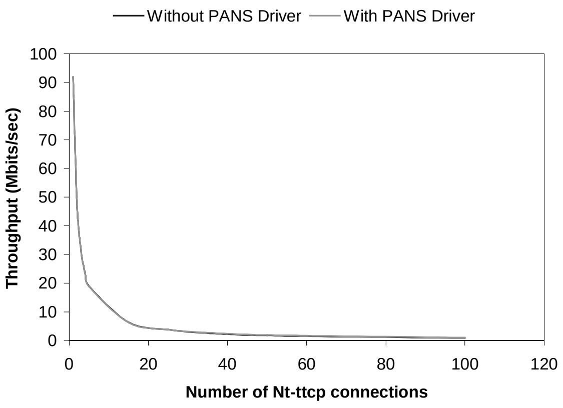


**Figure 6**: Throughput of the PANS Verifier using `ntttcp`

[7] If the verifier is implemented in the APs hardware, the AP may identify the MAC address of the attacker and filter against it.

each 64K in size. We repeated the execution of ntttcp by varying the number of threads (simultaneous connections between sender and receiver). Each time we measured the total throughput as recorded by the receiver machine. Since the parameters that we used for buffer size and buffer count were sufficient to flood a 100 Mbit/sec link, we were actually measuring the net throughput of the PANS Verifier. The entire run was first conducted without the PANS intermediate driver installed on the server and client machines. During the second run, both machines had the PANS driver installed. In addition, we ran a performance monitor on the PANS Verifier to record in real-time the average CPU utilization for the duration of each run of ntttcp. Our results are shown in **Figure 6** and **Figure 7**. While there is no significant difference in the throughput with and without the PANS intermediate driver, we see an 40% difference in the average CPU utilization of the PANS Verifier between the two situations.

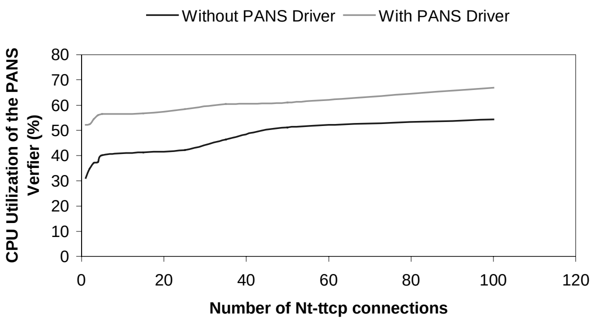


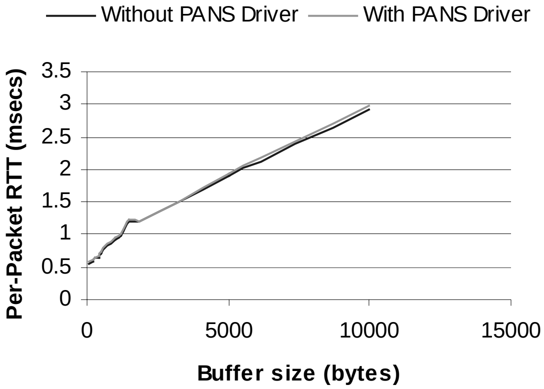
**Figure 7**: The CPU Utilization of the PANS Verifier as a function of the number of ntttcp connections. On average, the CPU utilization increases by 40% in the presence of the PANS Intermediate driver.

Another experiment that we performed was PANS with PPTP [19]. This simulated the case where individuals access their company's internal network by creating a VPN from a public network. PPTP is a protocol for creating such VPNs. We noticed that with PPTP the average throughput per connection went down by as much as 50%. For example, for 10 simultaneous connections the average throughput per connection went down to about 4.35 Mb/sec and for 30 simultaneous connections to about 1.02 Mb/sec. This was with compression and data encryption, which are part of the PPTP protocol, disabled. We noticed that with PANS the change in throughput was minimal, i.e. PANS added very little overhead to the overall throughput.

### 5.3 Effect of PANS Verifier on Packet RTT

For round-trip time measurements we used tcpperf. We flooded the network with 100,000 buffers, varying the packet size during each run. The plot in **Figure 8** shows the variation of per-packet RTT with the buffer size sent. The RTT values were compared in the presence and absence of the PANS intermediate driver. From the plot it can be seen that the per-packet RTT difference is in the order of tens of microseconds, which is not a very significant.

Our experiments were conducted using the default buffer size, which simulates bulk data transfers. We noticed that the 100 Mb/sec backbone link gets saturated before the Verifier does. We measured the throughput of the 11 Mb/sec IEEE 802.11 wireless LAN AP from a couple of different vendors and found that the average throughput for these networks is around 6 Mb/sec. This means that at least for the case of bulk transfers where the wired network is the bottleneck instead of the CPU, one single Verifier can handle as many as 10 APs.



## 6 Related Work

We are aware of a few systems that address some of the problems that the CHOICE network tackles and survey them below. The focus of these has been organization or enterprise-centric focusing on user authentication to keep out unknown persons from a wired corporate or university network. We on the other-hand tackle the problem of network access from an individual-centric perspective, establishing the identity of a **previously unknown** user and then give her **wireless** access to the public network. While several proposals take a hardware-based approach to providing authenticated access, we instead have opted to take a software-based approach that works today, and provides enough flexibility within the system for the host organization to be creative in the type and number of services it offers to its users.

The only fully deployed system that we are aware of is the SPINACH system developed as part of the MosquitoNet project at Stanford University [25]. The campus contains several publicly available network ports that mobile users can connect to by authenticating themselves through SPINACH. The user login process in process in SPINACH is similar to that of CHOICE except that instead of a global authentication database, SPINACH uses the campus Kerberos server. The highlights of this system are its ease of use, the innovative reuse of existing infrastructure, and no additional software requirement in the client. Nevertheless, while the authentication in SPINACH is done using an end-to-end secure model, subsequent network transactions between the users and the network are not authenticated on a per-packet basis. SPINACH keeps a log of the (IP, MAC) address pair of every user who has been successfully authenticated and filters incoming packets based on this tuple. Unfortunately, this model does not protect against hardware address spoofing. The architects of SPINACH note that a way to avoid hardware spoofing attacks is to use a stronger mechanism like IPsec for authentication.

A system proposed at UC Berkeley, is built around an "*authenticated DHCP*" server and requires both software and hardware support for controlling access from network ports [20]. By dynamically enabling and disabling ports in a network switch/hub and "sensing" an active node, the network issues a routable IP address only to authenticated users. Using Kerberos, the DHCP server performs authentication and if that succeeds it gives out an IP address to the node. If authentication fails, it resets the LAN hub port to a non-forwarding state. In addition to requiring specialized hardware in the switches, this solution is not a viable option for authenticating wireless users whose presence or absence in the network cannot be *sensed*. As in the case of SPINACH, this design does not solve the hardware address-spoofing problem.

Another intelligent hub-based network access control approach (layer-2) proposed within the IEEE standards committees is described in [21]. Briefly, the approach taken is to encapsulate a Extensible Authentication Protocol (EAP) frame within the Ethernet frame (EAPoE). The network port allows EAP encapsulated Ethernet frames with a specific multicast address to go through, which are then forwarded to a network RADIUS server for authentication [23]. If authentication is successful, the user receives a key and gains access to the network. Again, implementing EAP-based authentication in an IEEE 802.11 network requires specialized hardware in the access points. Further, an attacker can replace the AP with its own rogue AP which can negotiates a lesser form of authentication in order to perpetrate a dictionary attack to recover the user's password. CHOICE solves all these problems using lightweight software that offers more flexibility and functionality. When the PANS Verifier is implemented in the AP's firmware, all access to the network can be stopped at the port of entry.

The CMU NetBar system is yet another proposal in which all NetBar ports are isolated on a "non-connected" VLAN [24]. When a user attaches to the network, she gets an IP address from a DHCP server located on the "non-connected" VLAN allowing her to authenticate to a server on that network with her username and password. If authenticated, the server communicates with the switch and moves her to an "attached" network with full network connectivity. The system relies on a specialized hardware switch and is not secure from hardware address spoofing attacks. Furthermore, compared to the CHOICE network it offers a small fraction of the functionality at a much larger cost.

A final piece of work that is relevant to the CHOICE network is the one proposed in [25] by Patel and Crowfort. In this paper, the authors argue that in the future the need for home location based service contracts will become unnecessary. Their solution is based on the notion of tickets, where tickets are a form of electronic payment. A user uses these tickets to get service from the local service provider when she is away from home. Likewise, CHOICE's individual-centric approach allows users to not worry about the ISP they have signed a contract with or the company they work for in order to access the Internet and other network services.

## 7 Discussions and Future Work

In order for the CHOICE network to succeed, there has to be a business model that supports it. We feel that the CHOICE network architecture is very compelling and that everyone involved with it can benefit from it. In particular, the end-user benefits because it gives her a viable choice in how she accesses the Internet from places other than her office and home with the option of choosing different levels of services. The hardware vendors benefit as they can sell more wireless hardware. The ISPs and the network providers benefit as their resources are bought and used. The building owners benefit as they can use Internet Access to attract more visitors or customers by offering value-added services. Finally, the software vendors benefit as they can sell new types of functionality over this network.

We have deployed the CHOICE network in a local mall (Crossroads Mall, Bellevue, Washington). Currently, this network lets visiting Microsoft employees equipped with an IEEE 802.11 wireless network card access the Internet after they are authenticated via MS Passport. Very soon we will be opening this network up to the general public offering them additional services. We are currently building local services such as access to public printers, splash screens of list of upcoming activities/event, in-store buddy lists, and location guides as part of this network. We are using this pilot to fine tune PANS and are using it to carry out research on connectivity and computing in public spaces.

## 8 Conclusions

In this paper we argue that 3G networks will be frustratingly slow and that high-speed wireless LANs deployed in public places is the way to go for Internet connectivity in the wide-area. We have described how this connectivity can be enabled with the CHOICE network and its underlying protocol, PANS. We have advocated an individual-centric approach, where previously unknown users are identified and granted secure access to the Internet using a high-speed wireless LAN. We have discussed design goals, system components, system implementation, and system operation for the CHOICE network. We evaluated PANS and showed that it is secure and scalable. We have compared our design to other documented alternatives and have made the business case for why the CHOICE network should be adopted and deployed. We are confident about our approach and the value of our system to the point that we have deployed this network in a local mall where we hope to exercise all the different aspects of our design.

## References

[1] ITU-R Rec. M. 1225, "Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000,"

[2] IEEE 802.11b/D3.0, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High Speed Physical Layer (PHY) Extensions in the 2.4 GHz Band, " 1999

[3] Aironet Wireless Communications Inc, "Developer's Reference Manual: PC4500/PC4800 PC card Wireless LAN Adapter,". 1999

[4] R. V. Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, "New High-Rate wireless LAN Standards," *IEEE* Communications Magazine, Vol. 37, no. 12, pp. 82-88, December 1999

[5] J. Lansford, and P. Bahl, "The Design and Implementation of HomeRF: A Radio–Frequency Wireless Networking Standard for the Connected Home," *Proceedings of the IEEE*, November 2000

[6] "Single Computer Breaks 40-bit RC4," January 10, 1996,http://www2.ecst.csuchico.edu/~atman/Crypto/misc/netscape-icebreaker.html

[7] R. Droms, "Dynamic Host Configuration Protocol," *IETF RFC 2131*, March 1997, http://www.ietf.org/rfc/ rfc2131.txt

[8] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de. Groot, "Address Allocation for Private Internets," *IETF RFC 1597*, March 1994, http://www.ietf.org/rfc/rfc1597.txt

[9] Active Server Pages: http://msdn.microsoft.com/workshop/ server/asp/ASPover.asp

[10] R. Atkinson, "Security Architecture for the Internet Protocol", *IETF RFC 2401*, November 1998, http://www.ietf.org/rfc/ rfc2401.txt

[11] MS Passport: http:/www.passport.com

[12] T. Elgamal, S. Cotter, and the Netscape Security Team, "Netscape Security: Open-standard Solutions for the Enterprise,1998", http://developer.netscape.com/docs/ manuals/security/scwp

[13] P G. Viscarola and W. A. Mason, Windows NT Device Driver Development, *OSR Open System Resources*, 1999

[14] National Bureau of Standards, "Data Encryption Standard" *FIPS PUB 46*, January 1977, and "DES Modes of Operation," *FIPS PUB 81*, December 1980

[15] M. Rosing, Implementing Elliptic Curve Cryptography, Manning Publication Co. 1998

[16] CCITT.Recommendation X.509: The Directory-Authentication Framework, Geneva, 1989

[17] VeriSign Inc., Internet Trust Service http://www.verisign.com/

[18] R. Stine, "FYI on a Network Management Tool CatalogTools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices," *IETF RFC 1147*, April 1990, http://www.ietf.org/rfc/rfc1147.txt

[19] K. Hamseh, G. S. Pall, W. Verthein, et. al., "Point to Point Tunneling Protocol (PPTP) Technical Specification, June 1996,http://infodeli.3com.com/infodeli/tools/remote/general/pptp/pptp.htm

[20] D. L. Wasley, "Authenticating Aperiodic Connections to the Campus Network," June 1996 http://www.ucop.edu/irc/wp/wp_Reports/wpr005/wpr005_Wasley.html

[21] *IEEE Draft P802.1x/D1*, "Port Based Network Access Control," September 1999

[22] G. Appenzeller, M. Roussopoulos, and M. Baker, "User-Friendly Access Control for Public Network Ports," *Proceedings of INFOCOM '99*, March 1999

[23] C. Rigney, A. C. Rubens, W. A. Simpson, S. Willens, "Remote Authentication Dial-In user Service (RADIUS)," IETF RFC 1238, http://www.ietf.org/rfc/rfc128.txt

[24] E. A. Napjus, "NetBar - Carnegie Mellon's Solution to Authenticated Access for Mobile Machines," CMU White Paper, http://www.net.cmu.edu/docs/arch/netbar.html

[25] B. Patel and J. Crowcroft, "Ticket Based Service for Mobile User," *Proceedings of MobiCom '97*, pp. 223-233, September 1997

[26] S. Kent and R. Atkinson, "IP Authentication Header", *IETF RFC 2402*, Nov. 1998, http://www.ietf.org/rfc/ rfc2402.txt

[27] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)", *IETF RFC 2406*, November 1998, http://www.ietf.org/rfc/ rfc2406.txt

[28] Harkins D., and D. Carrel, "The Internet Key Exchange (IKE)", *IETF RFC 2409*, November 1998, http://www.ietf.org/rfc/ rfc2409.txt

[29] Microsoft Virtual Private Networking (VPN) White Paper, http://www.microsoft.com/ntserver/commserv/deployment/planguides/VPNSecurity.asp

[30] The Wireless Application Protocol (WAP) White Paper, http://www.wapforum.org/what/whitepapers.htm