On Securing Wireless LANs and Supporting Nomadic Users with Microsoft's IPSec Implementation

Rodrigo Blanco Rincon, Günter Schäfer
Telecommunication Networks Group
Technische Universität Berlin
10587 Berlin, Germany
[blanco, schaefer]@ft.ee.tu-berlin.de

Abstract

Wireless LANs, like the IEEE 802.11 WLANs, are more vulnerable than their wired counterparts. The IEEE 802.11 specification includes an encryption protocol, WEP (Wired Equivalent Protocol), but this protocol inhibits severe weaknesses: there is no automatic key distribution protocol and WEP's security itself has been shown to be seriously flawed. As a result, many of today's IEEE 802.11 networks are relatively easy for outside attackers to break into.

Predictions point at the fact that home and small to medium-sized office WLAN environments will be of great importance in the near future of the wireless market. A security system taylored for them and their "average" users should include a series of particular features: strong security, simplicity of installation and use, password management policies, user roaming capabilities and no special software or hardware requirements.

The approach presented in this paper[1] *consists in building a Virtual Private Network (VPN) over the WLAN, using IPSec as underlying security protocol. The proposed configuration solution performs Mobile Node authentication, automatic IPSec policy configuration and automatic generation of IPsec authentication keys (IKE's "Preshared Keys"). In order to support nomadic users, a policy negotiation protocol has been developed that allows to dynamically adjust the IPSec policies in mobile devices and the security gateway of a WLAN. The approach has been validated for the Windows 2000 / XP operating system with a prototypical implementation that is available for free download [2].*

1 Introduction

The security of 802.11 Wireless LANs still remains to be a problem: many WLANs are operated with little or no security at all. This is partly due to the IEEE 802.11 security mechanisms' limitations: the Wired Equivalent Privacy (WEP, the security solution integrated in WLANs) provides no key management and, even worse, severe vulnerabilities have been detected and implemented in attacking tools that are available for download in the Internet [3, 15, 16].

Since sensible data may be sent over wireless links, it is highly desirable to protect those links against eavesdroppers. Data modification or unauthorized access to services, among others, are further attacks that must be prevented.

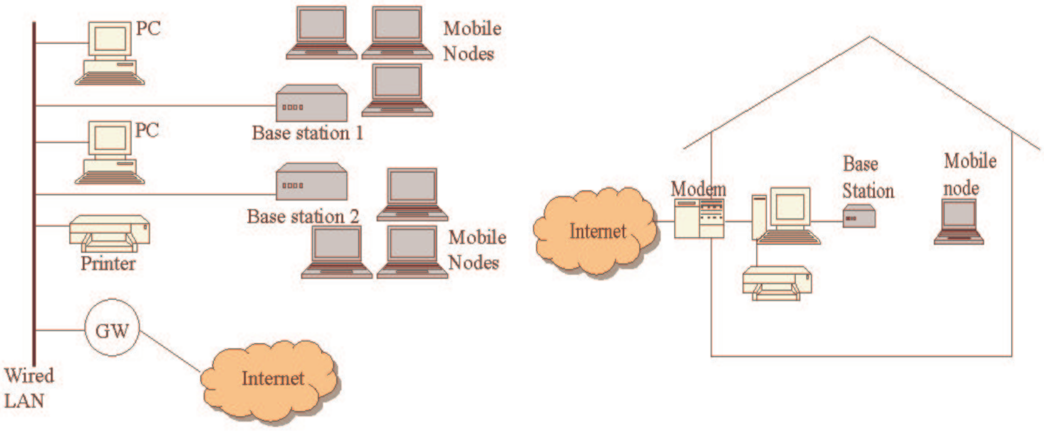
This paper describes a VPN-based configuration solution for an infrastructure WLAN [7] with the features of home and small to middle-sized office scenarios and a Windows 2000 Professional framework.

The usual scenario of roaming access to wireless LAN infrastructures is the following: a number of mobile stations (typically notebooks with wireless cards) get connected to the WLAN in order to access the corporate resources. These resources include Internet access, printers and other devices.

Two typical cases are taken into consideration in which these kinds of networks are used: in small and middle-sized corporate environments, and at home. Figure 1 shows both basic architectures.

Normally, there will be one or more Base Stations that transmit the information belonging to one or more WLANs. The base stations constitute the actual interface between the wired and wireless LANs: they are directly connected to the wired LAN (Ethernet) or the home computer with access to the Internet through some modem or network adapter. Note that the network resources and services such as printers or Internet access are shared among all users. Wireless users should also have access to these resources.

[1] This work has been supported with a grant from Micosoft Research, Cambridge, UK.



Office scenario

Home scenario

Figure 1. WLAN Scenarios for Home and Small to Middle-Sized Office Environments

In some cases, the home configuration may also include a small fixed network connecting a few PC's with a printer and a PC accessing the Internet. The PC wired to the Base Station probably has no wireless card.

In both scenarios, users need to connect their wireless-enabled notebooks to the wireless networks. In each of the wireless scenarios, users need to hold their communications securely. "Seamless" roaming, that is, that the change of WLAN, access point and wireless environment remains as transparent as possible to the users, together with homogeneous network security mechanisms, are desirable features. Users should also remain relatively unaware of the details regarding the underlying security mechanisms: they should be able to just turn on their notebooks, authenticate themselves and start working. However, they should be somehow aware that they have secured their communications.

2 Security Requirements and Desired Features

The proposed scenarios have a series of requirements in terms of security and software as well as hardware infrastructure. The security mechanisms must be usable for their average users, who typically need to access services beyond the WLAN limits. These issues are analyzed next.

The following security features must be assured in the wireless medium:

Access control: prevent unathorized Mobile Nodes from accessing services offered to the authorized users by the wired network or from talking to any entity belonging to the WLAN (authorized Mobile Nodes) or to the wired network (for example, wired workstations or the Internet). To accomplish this, an initial authentication step is needed. Entities that successfully go through the authentication process will be authorized to access the rest of the network entities and services. Non-authenticated entities can access neither the legitimate network entities (both wireless and wired) nor the services associated to the wired network.

Confidentiality: the wireless medium is much more easily accesible than wired networks. The information sent over the wireless links is much more prone to being eavesdropped on and therefore needs explicit protection. The confidentiality consists of encrypting the data flowing between entities with a key. The key management is the means by which the encryption keys reach the entities using confidentiality services. Only authorized entities should obtain valid keys. In that sense, it is convenient to associate the key management functionality to the user authentication process: once a user has authenticated himself, a key is generated and installed in the user's entity. Distributing individual keys (different for each user) is a safer approach as a "shared key" schemes, which have the following drawbacks:

- If an entity carrying the shared key is exposed, the data of all the other supposedly secured entites is also exposed
- In many cases, WLAN users will be expecting individual privacy: the traffic meant for an authenticated entity should remain undecipherable to the other authorized entities. This need is

present, for example, if guest users make use of the WLAN in a temporal and provisional way.

Data integrity / origin authentication: when an entity A receives a packet from another entity B, it can be sure that it is B that sent it and that nobody could change the contents of the packet without A noticing it. In other words, no entity different from B can send a packet to A impersonating B. Additionally, no entity can modify the data sent by B to A without A noticing it upon receival. Data integrity is carried out with an integrity check and the data origin authentication with a digital signature. A mixture of them can be found in the HMAC construct, which combines both of them [10].

Other desired features: Bearing in mind the average users of a WLAN, no advanced knowledge of networks or systems configuration should be expected from them. Instead, it is highly desirable to automate as much as possible the configuration steps that lead to the security goals enumerated above.

Additionally, no special hardware or software should be required: the configuration solution must run on standard hardware. It must interact with the standard software which is reasonably expected to be installed in the proposed environments, or software that can be freely downloaded and used.

Finally, users should have access to the services offered to the WLAN from entities beyond the wireless medium, that is, entities belonging to the wired network. These services include the use of diverse peripherals and access to Internet (which implies outwards IP visibility). They must also have the ability to communicate with other Mobile Nodes attached to the WLAN.

3 Overview of Technologies

3.1 IEEE 802.x technologies

There are two applicable technologies from the IEEE 802.x family: the IEEE 802.11's own security primitives and the IEEE 802.1x's authentication and key management capabilities.

The standard IEEE 802.11 [7] (for WLANs) includes some basic security services which are integrated in the WLAN environment: Shared Key authentication and Wired Equivalent Privacy (WEP). These inhibit some limitations:

- There is no key management mechanism to deliver the shared key to the participating entities within the IEEE 802.11 standard. The Shared Key is needed to accomplish the Shared Key authentication and the WEP encryption services.
- The shared key authentication can easily be circumvented by re-using keystream that can be re-covered from a previous run of the authentication protocol.
- Some weaknesses in relation with the integrity function and the reuse of the initializating vector in the WEP protocol were discovered and described in [3].
- In August 2001, a new attack to WEP was discovered, with which the shared key can be retrieved in less than 15 minutes provided that about 4 to 6 million packets have been recovered. The required effort grows only linearly with the number of bits used in the key, so using 40 or 104 bit keys (the two possibilities provided in the standard) makes virtually no difference at all. The weakness and the attack are described in [15] and [16]. This proves WEP to be insufficient to protect data flowing across WLANs.

IEEE 802.1x [8] is an IEEE standard approved in June 2001 that provides authentication and key management for IEEE 802 local area networks, including 802.11 WLANs. It does not provide encryption or encapsulation, and therefore adds no overhead to the packets. Its main purpose is to realize an access control to the ressources of a IEEE 802 LAN on the basis of an entity authentication dialogue.

Additionally, it may be used to solve the key delivery problem of the wired equivalent protocol (WEP), as it also supports negotiation and distribution of a session key during the authorization check.

However, WLAN environments using IEEE 802.1x keep relying on WEP as encryption and integrity protection protocol, which has been badly exposed. Another drawback is the possible limitations to upgrade the already existing wireless devices to this technology.

3.2 VPN technologies

Three VPN technologies are available for common operating systems: PPTP [5], L2TP/IPSec [4] and IPSec[1]. L2TP/IPSec consists on protecting L2TP with IPSec. A comparison of their properties is summarized in Table 1.

The authentication provided by PPTP and L2TP is user-based and happens only during the tunnel establishment. L2TP provides an additional authentication step. No packet data integrity / origin authentication is provided.

IPSec does not implement user authentication. Instead, it performs a host-based authentication. It also protects the packets with data integrity / origin authentication trailers, while PPTP or L2TP do not. IPSec's protocols to set up security associations (ISAKMP with IKE) [11, 6] is much more flexible than that of PPTP or L2TP.

The PPTP and L2TP tunnels are quite different to those of IPSec. PPTP and L2TP tunnels support dynamic configuration of their variables during the tunnel negotiation.

Table 1. Comparison of Available VPN Technologies

Aspect	*PPTP*	*L2TP/IPSec*	*IPSec*
Security services	*User-based authentication. No packet data origin / integrity authentication. PPP encryption. No replay protection*	*L2TP: User-based authentication. No packet data origin / integrity authentication. PPP encryption. No replay protection.* *IPSEC: Machine-based authentication (IKE). Packet data origin / integrity authentication. ESP encryption. Replay protection.*	*Machine-based authentication(IKE). Packet data origin / integrity authentication. ESP encryption. Replay protection.*
Tunnels	*Dynamic configuration of variables. TCP management.*	*L2TP: Dynamic configuration of variables. UDP management.* *IPSEC:Previous (static) configuration. No tunnel management.*	*Previous (static) configuration. No tunnel management.*
Key Management	*Initial key generation and periodic refreshment*	*L2TP: Initial key generation and periodic refreshment.* *IPSEC: IKE. Initial key generation and periodic refreshment*	*IKE. Initial key generation and periodic refreshment*
Multi-network	*PPP Payloads supported: IP, IPX, NetBEUI. IP-based PDU transport.*	*L2TP: PPP Payloads supported: IP, IPX, NetBEUI. Works in different network technologies: Frame Relay, ATM, X.25 and SONET.* *IPSEC: Payloads supported: IP. IP-based PDU transport.*	*Payloads supported: IP. IP-based PDU transport.*
Broadcast	*YES (with individual IP unicast packets)*	*YES (with individual IP unicast packets)*	*NO*
Overhead	*LOW*	*HIGH*	*Intermediate*
Level of Security	*LOW (no data integrity, no replay protection)*	*HIGH*	*HIGH*
Availability	*Windows, Linux, FreeBSD, Solaris, MacOS*	*Windows, Linux*	*Windows, Linux, FreeBSD, Solaris, MacOS, AIX*

They also need a continuous maintenance (implemented with a TCP connection in PPTP and a UDP protocol in L2TP). This requires both establishment time and bandwidth. IPSec tunnels perform no dynamic tunnel variable configuration.

The fact that L2TP tunnels can be built on different network technologies, such as Frame Relay, ATM or X.25 provides no advantage since in our scenario the underlying network is IP-based. Their ability to process different payload protocols (IPSec can not) is also irrelevant, since the only protocol considered is IP.

Multicast and broadcast traffic is protected by PPTP and L2TP. Microsoft's IPSec implementation, for example, claims to protect multicast traffic, but that does not make use of using multicast transmission in the WLAN. If IPSec is deployed, the broadcast packets would be unprotected. This is perhaps the only item in which PPTP and L2TP are clearly preferable to IPSec. However, there are some inherent drawbacks to the broadcast and multicast support in PPTP and L2TP as they must map multicast and broadcast packets to individual unicast packets for every receiver which may turn out to consume too much bandwidth in wireless LANs.

The performance is quite a problematic feature to compare. In terms of packet overhead, L2TP with IPSec introduces the biggest overhead and PPTP introduces the smallest overhead. More precisely, assuming that no IP header additional options are used, that the PPP header and padding have the maximal length (10 bytes and 4 bytes[2] respectively), that the IPSec padding has a maximal length of 8 bytes (3DES encryption is considered) and that no HMAC is added to the IPSec packets (since in PPTP no packet authentication is present): PPTP introduces 70 bytes, L2TP/IPSec introduces 100 bytes and IPSec introduces 78 bytes of overhead. A comparison of the protocol overheads is depicted in Figure 2. Additionally, PPTP and L2TP support payload compression, and L2TP also header compres-

[2] in order to align to the GRE packet

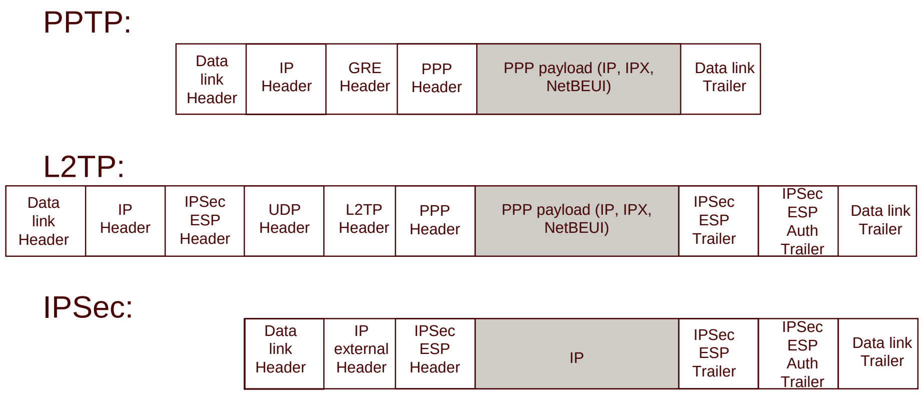


Figure 2. Comparison of the Different Protocol Overhead Structures

sion (under certain circumstances). In IPSec, compression can also be performed with the Payload Compression Protocol (PCP). L2TP and PPTP need extra control traffic for tunnel maintenance. IPSec introduces an intermediate overhead and requires no tunnel maintenance. In this sense, IPSec seems to be the preferable option.

PPTP is discarded a priori, due to its security flaws [13][14]. It is recommended that L2TP uses some lower-level protection, such as IPSec [17][4], due to some limitations, for example the limited protection of the L2TP tunnels. Furthermore, raw L2TP is not available in the Windows 2000/XP operating systems, which is regarded as an important requirement as most hosts are run with Windows software.

The choice would be then between IPSec and L2TP/IPSec. L2TP/IPSec introduces more overhead than IPSec and it requires tunnel maintenance, which is performed through a UDP-based management protocol. These two features hint at bandwidth requirements that could have a certain adverse effect over the WLAN bandwith-limited performance. The additional authentication, tunnel establishment and tunneling of L2TP appear to be redundant, since IPSec also provides them. L2TP/IPSec also provides the ability of carrying payloads different to IP, but in our scenario that is irrelevant, since the payload will be solely IP. Interoperability with other platforms might be an issue in the future. In this sense, IPSec is available in virtually all of the relevant operating systems, while L2TP/IPSec is not [12].

IPSec fits better the proposed environments' features and needs, and it provides a stronger and more flexible security solution. Therefore, it has been adopted as the VPN technology to protect the WLAN environments.

4 Proposed Topology for the WLAN Scenarios

In the WLAN scenarios, the interface between the wireless and wired network will be a workstation, which will be refered to as Security Gateway in this paper. It is a dual homed host that acts as the router between the mobile nodes and the wired network infrastructure. The Base Station[3] is directly connected to the Security Gateway's interface (Ethernet adapter) to the WLAN IP subnetwork. This is depicted in Figure 3.

A Security Domain is a wireless environment managed by a single Security Gateway. However, one Security Gateway could manage more than one Security Domain, so that what really defines the limits of a certain Security Domain is the relationship between Mobile Nodes' names and passwords (namespace). A Security Domain is limited to a C-class IP subnetwork's address range (network mask 255.255.255.0, 255 addresses). For each additional network interface (adapter) installed in the Security Gateway, another C-class IP subnetwork allows a new Security Domain to be supported.

Summing up, each network interface defines a new C-class IP subnetwork. This IP subnetwork supports a Security Domain in which the Mobile Nodes' identities and authentication credentials are independent to those valid for the rest of the Security Domains. In a Security Domain there is a theoretical maximum number of 253 Mobile Nodes (which is an acceptable number for small and medium-size environments). In most cases a single Base Station would normally not manage more than ten active Mobile Nodes at the same time, because of bandwidth constraints of the wireless link.

Although in Figure 3 only one Base Station has been

[3] It could happen that there is more than one Base Station.

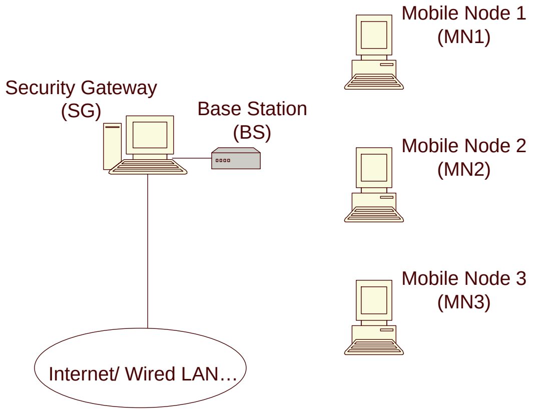


Figure 3. Topology for WLAN Scenarios

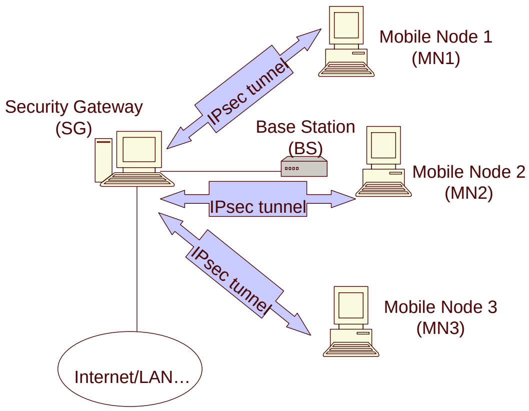


Figure 4. Scenario and Tunnel Configuration for IPSec Protection of WLAN Traffic

drawn, it would be possible to install more than one Base Station if, for example, the physical extension of the Security Domain requires it. The solution is to connect a hub/switch to the Security Gateway's IP interface for the Security Domain. Different Base Stations could be then connected to the hub/switch.

5 Architecture of the WLAN for IPSec security

Once IPSec has been chosen as security technology, some design decisions must be taken with regard to the network configuration and IPSec options. The IPSec protocol (AH or ESP), the IPSec mode (transport or tunnel) and the authentication method, among other issues, will be studied in this section.

IPSec protocol IPSec supports 2 protocols: AH and ESP. AH (Authentication Header) provides data origin authentication and replay protection. ESP (Encapsulating Security Payload) provides data origin authentication, confidentiality and replay protection. The choice is obvious: only ESP can protect data from malicious eavesdropping, since AH does not include encryption (confidentiality). 3DES is selected as encryption algorithm and SHA-1 as hash algorithm.

IPSec mode IPSec can be operated in two modes: transport mode (when the "cryptographic endpoints" coincide with the "communication endpoints" of the secured IP packets) and tunnel mode (when at least one of the "cryptographic endpoints" is not a "communication endpoint"). In our case, the cryptographic endpoints will not normally concur with the communication endpoints: the cryptographic associations will occur among members of the WLAN, but no further. In many cases, the Mobile Nodes will be communicating with entities outside the WLAN (through the wired network) that may not know about IPsec. For this reason, tunnel mode will be used.

The following question arises: between what entities should the IPsec tunnels be built? Can any Mobile Node build an IPsec tunnel to any other Mobile Node? Of course, this would be a possibility. But it would complicate too much the WLAN security client entities running on the Mobile Nodes. Just imagine that every node needs to be registered in every Mobile Node in order to be able to talk to it.

The tunnel partner of the Mobile Nodes must be able to perform a mutual authentication with them. It is much more sensible to centralize this complexity in a single workstation, the Security Gateway, which acts as a WLAN security server. All the tunnels are established between the Security Gateway and the Mobile Nodes, who need not know about the other WLAN Mobile Nodes. This way, the information about the different WLAN Mobile Nodes is centralized, and it is not so difficult to update client data, add/remove users, etc. This can be viewed in Figure 4.

IKE authentication In IPsec, the IKE authentication step, which is intended to negotiate the IPSec Security Association, can be performed using one of these three methods: Preshared Keys, Certificates and Kerberos. Home and small to medium-sized enterprise environments are unlikely to be deploying Kerberos[4], so this option has not been pursuit.

Using digital certificates would be an elegant solution to the authentication problem. The users, as well as the Secu-

[4] After all, no special or sophisticated software components must be required in this solution

rity Gateway, would produce an RSA key pair. A WLAN-local Certification Authority (CA) would sign the users' Public Keys, producing WLAN-local user certificates. With these signed Public Keys, their Private Keys and the Public Key of the local Certification Authority, users would produce Windows PKCS12 certificates and place them in the corresponding IPsec policies. In the proposed scenarios, this approach has a number of implications:

- The certificates used by the Security Gateway and *all* of the users for the IKE authentication are signed by the same CA. This implies that users trust not only the Security Gateway, but also *every* entity signed by the local Certification Authority, namely the other users. Essentially, the trust relationship should be restricted to each Security Gateway - Mobile Node pair. If the security of an entity registered under a certain Security Domain is exposed, this would imply that some attacker might have access to an entity that the other users of that Security Domain *trust*.

- If some client's certificate is exposed, the only solution in order to avoid endangering the rest of the users registered under the same Security Domain would be to enter the exposed certificate in a revocation list. However, this solution is not very convincing: in order to avoid accepting such certificates, the entity would need access to some server where the revocation list is avalilable. In our scenario, this cannot be expected: when the IKE negotiation takes place, users have no Internet access. Another possibility would be to produce a new CA in the affected Security Domain, and renew *all* the users' certificates. Their old certificates would need to be removed from their machines and the new ones would have to be installed. In either case, this solution is inefficient.

- CAs are generally operated under rigid security conditions. This kind of conditions cannot be expected from home and small to middle-sized environments. Operating a CA under poor security conditions is not to be recommended.

These implications prove that using certificates for the IKE authentication step does not suit the proposed scenarios' needs.

So the only option left is Preshared Keys. From this standpoint, a Preshared Key (a simple string which matches on both IPsec partners) will be kept on both the Security Gateway and the Mobile Nodes, or created dynamically every time they want to establish an IPsec association. Of course, the Preshared Key of the Security Gateway with each Mobile Node is different. The Preshared Key of every Mobile Node with each Security Gateway is also different.

In the proposed environments, the Preshared Keys for every pair Mobile Node - Security Gateway are generated dynamically and refreshed for every new session, as explained in Section 6.

Configuration files containing information (including the Preshared Secrets) about the registered peer entities (Mobile Nodes for the Security Gateway, Security Domains for the Mobile Nodes) must be kept on every Mobile Node and Security Gateway.

Structure of the IPSec-secured WLAN The WLAN environment secured with IPsec will have a number of particular characteristics. The Mobile Nodes communicate solely with the Security Gateway, which decides if the packets should be forwarded and in what direction. So, in fact, the network segment to be protected encompasses the connections of all the Mobile Nodes with the Security Gateway.

Every Mobile Node establishes an IPSec tunnel with a Security Gateway, which acts as the IPsec association counterpart for every mobile node. This must be accomplished whenever a Mobile Node roams into a new Security Domain. That means that the IPSec policy for both the Mobile Node and the Security Gateway must be dynamically updated whenever a new entity roams into a new Security Domain. This can be viewed in Figure 4. This tunnel is the product of a Mobile Node authentication protocol run, in which the Mobile Node and the Security Gateway negotiate dynamically the tunnel configuration parameters. This protocol has a double functionality: Mobile Node and Security Gateway mutual authentication and generation of a session IPSec Preshared Key for the IPSec tunnel.

If a mobile node wants to establish a communication with an IP address of the wired infrastructure or the Internet, it needs to send its packets through the IPsec tunnel to the Security Gateway, which will route them outside. If it wants to talk to another node in the WLAN, it must first send his packets to the Security Gateway (through the IPsec tunnel), who will forward the packets to the other mobile node. Thus, the exposed part of the network, that is, the wireless links, is protected.[5].

The Mobile Nodes only accept traffic coming from the Security Gateway, through the IPSec tunnel. This way, non-authorized entities cannot access them: if they try to talk directly to the authorized (and properly configured Mobile Nodes), their packets will be blocked by the legitimate entities. If they try to talk to entities belonging to the wired network or to the Security Gateway, their packets will be blocked (unless they are tunnel negotiation protocol traffic). Unfortunately, two unconfigured Mobile Nodes could

[5] This is why IPsec's tunnel mode is required: one part of the IPsec association must be obligatorily the Security Gateway, and at the same time, the Security Gateway does not necessarily have to be one of the communication endpoints.

still talk to each other using the WLAN's bandwidth, since no access control can be performed in the Base Station on the basis of IPSec.

Further configuration Up to now, IPSec seems to fulfil fairly well the proposed scenarios' security needs: it provides packet data integrity and origin authentication. It also performs an initial IKE authentication (entity-based) in order to set up the IPSec Security Associations [6], based on some information generated independently of IPSec. However the configuration of the IPSec policies is assumed to have been performed by hand before the tunnel can be established. That implies that some additional authentication must be provided somehow, in order to setup the tunnel configuration and the authentication material (Preshared Key) used in the IKE authentication.

Thus, some functionality must be added to automate the dynamic modification of IPSec policy that is required whenever the user roams into another security domain.

6 Prototypical Implementation for Dynamic Configuration of IPSec Policies

In the previous sections, a description of the typical features of small and middle-sized WLAN environments has been given. The most relevant security technologies available to protect these wireless scenarios have been considered, and IPSec has been chosen as the best option to build a VPN over the IEEE 802.11 WLANs. The appropriate IPSec options for the proposed environments have also been analyzed.

We implemented our prototype for the Windows 2000/XP operating system because of two main reasons: because IPSec is integrated in these operating systems, and because the majority of workstations and mobile nodes up to date is running a Windows operating system.

In order to deploy IPSec in the wireless scenarios without expecting specific knowledge from the users, some functionality must be added: before IPSec tunnels can be established in the WLAN, some configuration steps in the participating entities must be accomplished. The issues which must be addressed are:

- Roaming: since the IPSec policy must be configured before the communications are secured by IPSec, every time a user roams into a new Security Domain it is necessary to update the IPSec policies both in his Mobile Node and in the Security Gateway. To perform these operations by hand is not convenient: it requires specific IPSec knowledge from the users and the process is too slow and not scalable. Thus, it is preferable to establish a means to accomplish these configuration tasks automatically.
- Authentication: the Preshared Keys for every pair Mobile Node - Security Gateway are generated dynamically and refreshed for every new Security Domain in which the Mobile Node roams. Some threats explained in Section 7 point at the need for a Mobile Node and Security Gateway mutual authentication and IPSec policy negotiation protocol which must be implemented independently from IPSec. The Preshared Keys are derived from two kinds of information:
    - Random material: random information is exchanged between the peer entities in order to provide some randomness and unpredictability to the keys.
    - Preshared secret: a pre-shared secret is present in both communicating entities (Mobile Node and Security Gateway) for the purpose of negotiating an IPSec policy for each Mobile Node - Security Gateway pair, before the negotiation protocol takes place. From it, they can derive the same IPSec Preshared Key without sending the authentication-sensible information (the preshared secret) though the wireless link. It can be seen as the Mobile Node's "password" in a certain Security Domain.

The entities taking part in this WLAN architecture need some previous configuration before their communications are secured with IPSec. The necessary previous configuration can be classified in three categories:

1. Basic configuration: it must be accomplished only once, before any other configuration takes place.
2. Registration in a new Security Domain: performed after the manual configuration, every time a Mobile Node needs to be registered in a new Security Domain.
3. Dynamic IPSec policy configuration: only after the manual and initial configuration have been completed; it takes place every time a user enters another Security Domain and wants to negotiate an IPSec tunnel with the correspondong Security Gateway.

In the following sections, the description of the configuration steps refers partly to a Windows 2000/XP environment, and the names of the software components (that are part of the developped prototype) involved in each configuration step are written in italics.

6.1 Basic configuration

The configuration steps described in this section must be accomplished only once, after the installation of the software and before any other configuration is performed. They are intended to enable some necessary settings in the Security Gateway and each Mobile Node, as well as to prepare the entities for the tunnel negotiation protocol.

In the case of the Security Gateway:

- Network Address Translation, NAT (if deployed): it is a good practice (although not compulsory) to use private IP addresses in the WLAN. If private addresses are used, NAT is needed in order to provide the Mobile Nodes with IP visibility.

- DHCP Server (if deployed): it is useful to distribute the IP addresses to the Mobile Nodes dynamically as they enter the WLAN. It is not compulsory to install it.

- IP forwarding (*EnableRouting.reg*): it is necessary if the Mobile Nodes need outbound IP visibility (to the wired network and the Internet), which is the general case. In the Windows 2000/XP operating system, for example, this can be achieved by changing a value in the Windows registry or by executing a "registry file" which is provided with the software bundle (making changes directly in the Windows registry is an error-prone practice, and it is not recommended for the average users).

- Security Gateway ID (*SGNameConfigurator.exe*): the Security Domain identifier is the name of the Security Gateway. In order to avoid Security Domain name collisions, the Security Gateway's name is generated by concatenating its hostname with a number of randomly generated bytes. The Security Gateway identifier is saved in a special file. It must not be changed.

- Random Seed Source (*RandomInit.exe*): this stores in the Security Gateway some random information derived from the Security Gateway's administrator random keystrokes. This is later used to generate good quality (highly unpredictable) random pieces of information for the IPSec tunnel negotiation protocol.

- Initial IPSec blocking rules (*InitialIPSecConfigurator.exe*): this installs an initial IPSec protection in the Security Gateway, so that Mobile Nodes can access the legitimate network entities (both wired and wireless) only after performing a successful authentication and IPSec tunnel negotiation protocol.

In the case of Mobile Nodes:

- DHCP client (if deployed): it is useful to obtain the IP address for the Mobile Nodes dynamically and automatically as they enter the WLAN. It is not compulsory to install it.

- Random Seed Source (*RandomInit.exe*): this stores in the Mobile Node some random information derived from the Mobile Node's user's random keystrokes. This is later used to generate good quality (highly unpredictable) random pieces of information for the IPSec tunnel negotiation protocol.

6.2 Registration in a new Security Domain

This configuration step takes place each time a Mobile Node needs to be registered in a new Security Domain. As pointed out in Section 5, the entities taking part in the IPSec tunnel protocol (the Mobile Nodes and the Security Gateway) need to keep a configuration file with the information for the negotiation protocol. In the case of the Mobile Node, a database with the different Security Domains' names and preshared secrets with the Security Gateways in which it is registered must be present. In the Security Gateway, a database with the different registered Mobile Nodes' names and the preshared secrets with each of them is also mandatory. Both databases must be updated when a new Mobile Node needs to be registered in a Security Domain.

In the case of the Security Gateway:

- Users' database (*server.conf*): a new entry is added, containing the new Mobile Node's name and the preshared secret with it.

In the case of Mobile Nodes:

- Security Domains' database (*client.conf*): a new entry is added, containing the new Security Domain's name and the preshared secret with its Security Gateway.

6.3 Dynamic IPSec policy configuration

When a Mobile Node enters a new Security Domain, it must run the authentication and IPSec tunnel negotiation protocol. This protocol allows the mutual authentication of the Mobile Node and the Security Gateway of the Security Domain. It also derives a fresh session IPSec Preshared Key from the preshared secret between the Mobile Node and the Security Gateway and some dynamically generated random material. It is assumed that the previous configuration steps have been completed.

Two entities take part in the protocol: the Mobile Node (running the *WLANClient.exe* application), acting as client, and the Security Gateway (running the *WLANServer.exe* application), acting as server. In order to start the protocol, the Mobile Nodes run their client applications, WLANClient:

1. The WLANClient application sends a request packet to the Security Gateway with the following information:
   - its hostname, *MN* (that is, the client's name)

Table 2. Notation of the Tunnel Negotiation Protocol

Notation	Meaning
MN	Identifier of the Mobile Node
SG	Identifier of the Security Gateway
IP_{MN}	IP address of the Mobile Node
IP_{SG}	IP address of the Security Gateway
r_{MN}	Random number (challenge) generated by the Mobile Node
r_{SG}	Random number (challenge) generated by the Security Gateway
SGN_{MN}	Signature of the Mobile Node over frame 3
SGN_{SG}	Signature of the Security Gateway over frame 2

- its IP address, IP_{MN} (recently acquired for this Security Domain)
- a random number generated by the WLANClient, r_{MN}

The Security Gateway is assumed to be located in the IP address given by the Default Gateway IP setting of the Mobile Node. The reason for this is obvious: the Security Gateway is, anyway, the default gateway for every Mobile Node.

$$MN \rightarrow SG: \quad \left(1, MN, IP_{MN}, r_{MN}\right)$$

2. The Security Gateway's server, WLANServer receives the request. If the Mobile Node is not registered in the Security Gateway's database, it sends back an error frame. Otherwise, the WLANServer goes on with the protocol. As a reply, it sends a packet to the WLANClient with the following information:

   - The Security Gateway's name, SG (which is the Security Domain's name)
   - The Security Gateway's IP address IP_{SG}
   - The Mobile Node's name MN
   - The Mobile Node's IP address IP_{MN}
   - a random number generated by the WLANServer r_{SG}
   - the random number generated by the WLANClient r_{MN}
   - A HMAC signature of all this information. SGN_{SG}

   This HMAC uses, among other things, the pre-shared secret between the Security Gateway and the Mobile Node. This pre-shared secret has the same function as a Mobile Node's password (machine-based authentication) in the Security Domain.

$$SG \rightarrow MN: \quad \left(2, SG, IP_{SG}, MN, IP_{MN}, r_{SG}, r_{MN}, SGN_{SG}\right)$$

3. The WLANClient application receives the reply. If the Security Gateway is not registered in its database, it sends an "abort" message. Otherwise, it reproduces itself the HMAC signature over the packet information (it can do so because it also has the pre-shared secret that was used by the WLANServer to sign the frame). If its signature matches the signature attached in the packet, the information is assumed to be authentic (the entity that sent the reply necessarily knows the pre-shared secret between the Mobile Node and the Security Gateway). Only the Security Gateway is able to produce a correct signature since only it knows this pre-shared secret. Hence, if the signature is valid, the peer entity is assumed to be the legitimate Security Gateway.

   If the signature was not authentic, the frame is dropped. Otherwise, the WLANClient also sends a confirmation message to the WLANServer. This confirmation states that the Mobile Node accepts the identity of the Security Gateway. Basically, it contains the same information as the WLANServer's reply, but signed by the WLANClient.

$$MN \rightarrow SG: \quad \left(3, MN, IP_{MN}, SG, IP_{SG}, r_{MN}, r_{SG}, SGN_{MN}\right)$$

4. As the WLANServer receives the confirmation from the WLANClient, it does the same as its counterpart: it reproduces itself the HMAC signature over the packet information. If its signature matches the signature attached in the packet, the information is assumed to be authentic (the entity that sent the reply knows the pre-shared secret between the Mobile Node and the Security Gateway). Apart from the Security Gateway, no other entity knows the pre-shared secret. Hence, only the Mobile Node is able to produce a correct signature. If the signature is valid, the peer entity is assumed to be the Mobile Node.

If the node's signature is not authentic, the packet is dropped. Otherwise, the WLANServer updates its IPsec policy, adding rules for an encrypted tunnel allowing the normal traffic between the Mobile Node and any other IP address to flow only through the Security Gateway. This tunnel uses Preshared Key (SK) authentication.

The Preshared Key (SK) for the IPsec tunnel authentication is dynamically produced by the WLANServer. It is the result of an HMAC using the pre-shared secret between the Security Gateway and the Mobile Node (the node's password for the present Security Domain) as key, and the random numbers generated by the Mobile Node and the Security Gateway and some constant value, which is fixed for every negotiation and part of the protocol specification, as data.

Finally, the WLANServer entity sends the WLANClient's confirmation (code = 3) back to it with code = 4. This packet does not need to be signed, since both parties have already been mutually authenticated. It does not provide additional information that needs to be signed in order to check the sender's identity. It just acknowledges that the WLANServer has already updated the IPSec policy in the Security Gateway.

$$SG \rightarrow MN : \quad \left(4, MN, IP_{MN}, SG, IP_{SG}, r_{MN}, r_{SG}\right)$$

5. Upon receipt of this packet, the WLANClient updates its IPsec settings. The settings include the rules for an encrypted tunnel allowing the normal traffic between the Mobile Node and any other IP address to flow only through the Security Gateway. This rule uses Preshared Key (SK) authentication. These rules are established between the Mobile Node and the Security Gateway, so the IPsec protection of the communications only covers this segment. The Preshared Key (SK) for the IPsec authentication is dynamically produced, following the same procedure as that explained above for the Security Gateway.

   Finally, the WLANClient entity launches a ping to the Security Gateway, in order to trigger the IPSec association setup between the Security Gateway and the Mobile Node.

This handshake protocol is suited to be run over an untrusted medium, since the preshared secret between the Mobile Node and the Security Gateway is never sent over the wireless link. However, the pre-shared secret itself (the node's password) can become a vulnerability: the simpler it is, the weaker the IPsec tunnel is. That is why users should choose long, difficult to figure out passwords[6].

7 Security analysis of the dynamic IPSec policy negotiation protocol.

Need for mutual authentication through the signed messages It could be considered to simplify the above mentioned authentication and IPSec policy negotiation protocol, by removing the mutual authentication. If only the Security Gateway and each Mobile Node know their shared secret, why to perform an authentication step? No authentication should be required since the IKE Preshared Key does not necessarily have to be dynamically negotiated (both peers have a preshared secret from which it is possible to derive the IKE Preshared Key). Only registered users would be able to build legitimate tunnels with the Security Gateway, since only they can produce the necessary IKE Preshared Key. The only information that the Security Gateway would need is the pair Mobile Node's name - present Mobile Node's IP address for each incoming Mobile Node. All the Mobile Nodes would need to know is in which Security Domain they have roamed.

This approach results in a lighter message exchange: when Mobile Nodes roam into a new Security Domain (in which they were already registered), they would just need to send an unauthenticated request to the Security Gateway, who would update its IPSec policy considering the Mobile Node's name and IP address, and deriving the IKE Preshared Key from the preshared secret saved in the Security Domain local users' file. It would then send an unauthenticated reply to the Mobile Node stating its identity and acknowledging that it has updated its IPSec policy. From the Security Gateway's identity, the Mobile Node would retrieve the corresponding preshared secret and derive the Preshared Key from its local file, and update its IPSec policy.

In that design, the protocol is just meant to let the Security Gateway know the identity of the arriving Mobile Nodes (so that it can dynamically update its IPSec policy to build a tunnel using the adequate IKE Preshared Key with them) and to let the Mobile Nodes in which Security Domain they are in (so that they can dynamically update their IPSec policy to build a tunnel using the adequate IKE Preshared Key with the Security Gateway). This implies a series of security weaknesses that advocate for the stronger negotiation (which performs mutual entity authentication) described before:

- Client DoS: If the Security Gateway reply messages are not authenticated, an attacker could impersonate the Security Gateway, declaring that Mobile Nodes

[6] For this reason, it is recommended to use a strong random password generator. More precisely, what is needed is high-enthropy passwords. Please note that passwords do not need to be easy to remember since the user will write them only once in his WLANClient configuration file, not every time he runs the WLANClient

have entered a different Security Domain from the actual one. Two things could happen: if the users do not know the Security Domain whose identifier is being delivered by the attacker, they would be unable to perform the IPSec configuration. If they know the Security Domain whose identifier is being delivered by the attacker, they would be able to configure their IPSec policy, but they would still be unable to establish the IPSec tunnels with the legitimate Security Gateway, since the Preshared Key they have entered in their IPSec policies corresponds to another Security Domain.

- Security Gateway DoS 1: a Mobile Node is identified in a Security Domain by its hostname. To configure its IPSec policy, the Security Gateway needs also the present IP address of the requesting Mobile Node in the Security Domain (which is a dynamic information) and the corresponding Preshared Key (which is derived from static data stored in a file). An attacker could issue malicious requests with arbitrary names and IP addresses. Since no authentication of the messages is performed, this would force the Security Gateway to change the correct PKE Preshared Key of "logged-in" users for other arbitrary users' Preshared Keys, making it impossible for the legitimate users to make use of the WLAN.

- Security Gateway DoS 2: in Windows, the means to automatically (that is, not manually) update the IPSec policy is by using the tool "ipsecpol.exe". This tool is very computation-demanding and the time needed to add or change rules grows with the number of rules in the active IPSec policy. In the proposed environments, the number of rules in the Security Gateway's IPSec policy is fairly high. An attacker could flood the Security Gateway with malicious requests. Since no mutual authentication is required, these non-authenticated requests are inconditionally accepted by the Security Gateway, who must update the IPSec policy for every incoming request. This would cause the legitimate WLAN users not to be able to "log in".

Nature of the preshared secret piece of data So, authentication in both senses has been proven to be necessary. Another question arises: although authentication is necessary, why having a preshared-secret in the participating entities instead of the IKE Preshared Key itself? In fact, it would be possible to store directly the IKE Preshared Keys of the Mobile Nodes with every Security Domain in a plain file in the Mobile Nodes and a plain file in the Security Gateway containing the Preshared Keys of the Security Gateway with every registered Mobile Node. This would avoid managing pre-shared secret information different from the IKE Preshared Keys: the Preshared Key (used for the IPSec's IKE Phase I authentication step) would be the only used piece of information.

Of course, the possibility is valid, but there is still a need for a preshared piece of information (which can be a preshared secret or the IKE Preshared Key itself) that can be used to sign the messages exchanged and to derive the IKE Preshared Key that is used when updating the IPSec policy both at the Mobile Nodes and at the Security Gateway. So the only real difference is that with the approach proposed above, the IKE Preshared Key (which is also used in the IKE authentication) is not used for signing both the policy negotiation messages as well as the IKE frames, but produced dynamically from the pre-shared secret and some other random nonces, which are different and unpredictable from one protocol run to the next. Since the IPSec policy negotiation protocol and the IKE are separate protocols, it seemed a cleaner approach to derive a different IKE Preshared Key from the pre-shared secret every time that a successful protocol run takes place. This way, two different keys are used for the two different protocols.

Mutual authentication support PKIX (entity certificates) might be used to sign the IPSec policy negotiation protocol messages instead of the proposed pre-shared secrets. Essentially, the mutual authentication would be equally valid as with the preshared secrets. The users, as well as the Security Gateway, would produce an RSA key pair. A WLAN-local Certification Authority (CA) would sign the users' Public Keys, producing WLAN-local user certificates. By using these certificates, the Mobile Nodes and the Security Gateway could also authenticate themselves. However, there is a drawback to this approach: CAs are generally operated under rigid security conditions. This kind of conditions cannot be expected from home and small to middle-sized environments. Operating a CA under poor security conditions is not to be recommended.

8 Conclusions

The IEEE's 802.11 Working Group is now developing a next-generation WEP, that is expected to be put into practice in two phases. For existing devices an intermediate solution has been developed that overcomes the cryptographic vulnerabilities of the WEP protocol with some hotfixes and that is supposed to be installed on existing hardware platforms via software upgrades. For longterm security an AES-based security protocol is being specified. As this solution requires hardware support for the AES-operations, however, it is not expected to be available for existing devices [9]. But even for the intermediate solution it will take some time until all vendors of already sold equipment can provide the required software upgrades. Therefore, alternative solutions like deploying VPN technology will still be required for quite a while.

VPN solutions were not originally intended to secure wireless environments. However, they are often assumed to be a valid solution to the WLAN security problems. This article discussed a study of how VPNs can be adapted to wireless environments. It can also be considered as an analysis of the present LAN and VPN technologies and their adaptability to small and medium-size WLANs.

Based on an analysis of the different security technologies available, IPSec has been proposed to secure the traffic of the legitimate WLAN users and protecting the associated wired infrastructure.

In order to support nomadic users, however, some software components had to be added to the WLAN entities to support dynamic negotiation of the IPSec security policies in mobile nodes and a WLAN's security gateway. With these extensions nomadic mobility for users roaming between multiple WLAN installations has been achieved. No special / expensive software or hardware is required to deploy this solution, as has been shown by a prototypical implementation for the Windows 2000 / XP operating system [2]. Complicated tasks such as the configuration of IPSec policies or updating of the Windows Registry have been automated so that the dynamic configuration of Mobile Nodes in roaming conditions remains simple and fast.

The approach fulfills the security demands of the considered WLAN scenarios (as described in Section 2): access control is enforced together with mutual entity authentication between Mobile Node and Security Gateway by the IPSec policy negotiation protocol and, in a second step, IPSec's IKE authentication. Confidentiality and data integrity / origin authentication are provided by individual IPSec encrypted tunnels established between each Mobile Node and the Security Gateway that manages the Security Domain.

The solution has, however, some limitations. IPSec does not protect multicast or broadcast traffic. This is an open issue that requires further analysis. Furthermore, its authentication is host-based: in a multi-user Mobile Node, there is no way to limit the access of each user to the WLAN IPSec tunnel. Finally, there may be interactions with other IPSec policy rules that have to be active on some mobile hosts. As IPSec has been designed for use in fixed networks, the dynamic policy updates required for supporting nomadic users have not been considered nor have the potential interactions between policy settings required for WLAN protection and policy settings required for proper VPN integration of devices. It is this last aspect, that gives us the impression that basic WLAN security should best be ensured in the WLAN protocol itself and not via higher-layer protocols.

References

[1] R. Atkinson and S. Kent. Security Architecture for the Internet Protocol. RFC 2401, 1998.

[2] Blanco, R. and Schäfer, G. SecureWLAN Home Page, Mar. 2002. `http://www-tkn.ee.tu-berlin.de/research/SecurityLab/securewlan_home_page_v2.htm`.

[3] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of ACM MobiCom*, 2001. `http://www.cs.berkeley.edu/~daw/papers/wep-mob01.ps`.

[4] P. et al. Securing L2TP using IPsec. RFC 3193, Nov. 2001.

[5] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-Point Tunneling Protocol (PPTP). RFC 2637, July 1999.

[6] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). Internet RFC 2409, 1998.

[7] Institute of Electrical and Electronics Engineers (IEEE). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. The Institute of Electrical and Electronics Engineers (IEEE), IEEE Std 802.11-1997, 1997.

[8] Institute of Electrical and Electronics Engineers (IEEE). Standards for Local and Metropolitan Are Networks: Standard for Port Based Network Access Control. The Institute of Electrical and Electronics Engineers (IEEE), IEEE Draft P802.1X/D11, 2001, 2001.

[9] Institute of Electrical and Electronics Engineers (IEEE). Proposed TGi D1.8 Clause 8 Editing Changes, Mar. 2002. `http://grouper.ieee.org/groups/802/11/Reports/tgi_update.htm`.

[10] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*, Feb. 1997. RFC 2104.

[11] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408, Nov. 1998.

[12] Microsoft Corporation. Virtual Private Networking. Windows XP Documentation, 2001.

[13] B. Schneier and Mudge. Cryptanalisys of Microsoft's Point-to-Point Tunneling Protocol (PPTP). *Proceedings of the 5th ACM Conference on Communications and Computer Security, ACM Press*, pages 132–141, 1998.

[14] B. Schneier, Mudge, and D. Wagner. Cryptanalisys of Microsoft's PPTP Authentication Extensions (MSCHAPv2). Counterpane Systems, 1999.

[15] A. Shamir, I. Mantin, and S. Fluhrer. Weaknesses in the Key Scheduling Algorithm for RC4, Aug. 2001. `http://eyetap.org/~rguerra/toronto2001/rc4_ksaproc.pdf`.

[16] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the Fluhrer, Mantin and Shamir attack to break the WEP, Aug. 2001. `http://www.cs.rice.edu/~astubble/wep`.

[17] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer Two Tunneling Protocol "L2TP". RFC 2661, Aug. 1999.