

Secure WLAN Operation and Deployment in Home and Small to Medium Size Office Environments

Dr. -Ing. Günter Schäfer, Rodrigo Blanco

March 6, 2002

Abstract

In IEEE 802.11 WLANs, the information is transmitted through the air, over a certain physical extension. This makes these networks more vulnerable than their wired counterparts. The IEEE 802.11 specification (for WLANs) includes an encryption protocol, WEP (Wired Equivalent Protocol), but it presents some weaknesses: there is no automatic key distribution protocol and the WEP's security itself has already been exposed. Up to date, IEEE 802.11 systems are relatively easy for outside attackers to break.

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 tailored 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 exposed in this paper consists in building a Virtual Private Network (VPN) over the WLAN, using IPSec as underlying security protocol. The configuration solution proposed performs Mobile Node authentication, automatic IPSec policies configuration and automatic IPSec session authentication keys (IKE's "Preshared Keys") generation.

1 Introduction

The security of 802.11 Wireless LANs 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, what is worse, it is no longer secure [1][10][11].

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

This paper describes a VPN-based configuration solution for an infrastructure WLAN [5] 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 this kind 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.

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 them, they need to hold their communications securely. "Seamless" roaming (the change of WLAN, access point and wireless environment remains to them as transparent as possible,

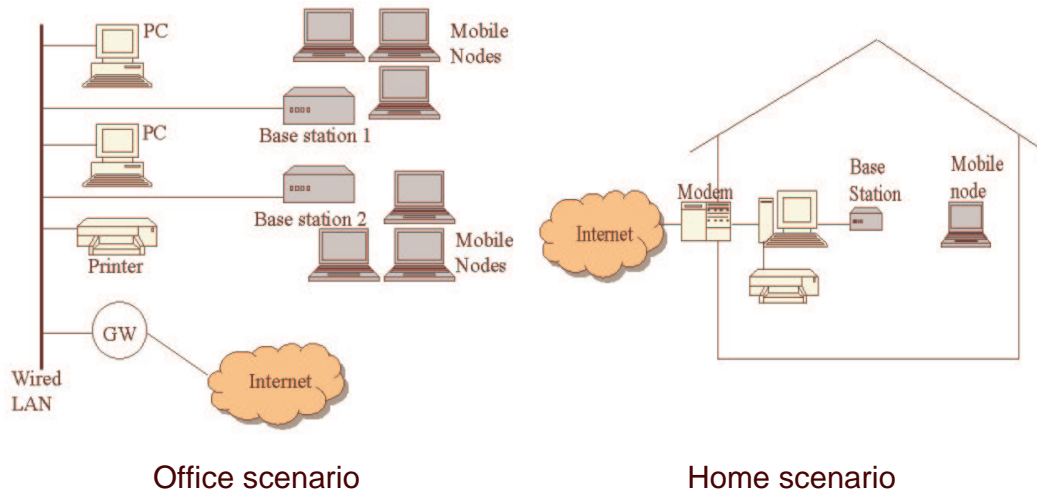


Figure 1: WLAN scenarios for Home and Small to Middle-Sized Office Environments

with homogeneous network security mechanisms) is a desirable feature. 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 conscious 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 and 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 (entity authentication): prevent unauthorized 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 accessible 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 entities 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 receipt. 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 powerful HMAC construct, which combines both of them [7].¹

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 automatize 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 normal 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 the other Mobile Nodes attached to the WLAN.

3 Overview of technologies and choice

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 [5] (for WLANs) includes some basic security services which are integrated in the WLAN environment: Shared Key authentication and Wired Equivalent Privacy (WEP). These present 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.
- An attack to WEP has been discovered, with which the network key can be retrieved in less than 15 minutes provided that about 4 to 6 million packets have been recovered. The required effort

¹HMAC was designed to cater for the authentication needs of the IPSec protocol.

Table 1: Comparison of the available VPN technologies

	<i>Authentication</i>	<i>Tunnels</i>	<i>Key Management</i>	<i>Multi-network</i>	<i>Broadcast</i>	<i>Overhead</i>	<i>Level of Security</i>	<i>Availability</i>
<i>PPTP</i>	<i>User-based. No packet data origin / integrity authentication</i>	<i>Dynamic configuration of variables. TCP management. NAT-compatible.</i>	<i>Initial key generation and periodic refreshment</i>	<i>NO</i>	<i>YES</i>	<i>LOW</i>	<i>LOW</i>	<i>Windows, Linux, FreeBSD</i>
<i>L2TP/IPSec</i>	<i>L2TP: User-based. No packet data origin / integrity authentication. IPSEC: Machine-based (IKE). Packet data origin / integrity authentication</i>	<i>L2TP: Dynamic configuration of variables. UDP management. NAT-compatible. IPSEC: Previous (static) configuration. No tunnel management. NAT-incompatible</i>	<i>L2TP: Initial key generation and periodic refreshment. IPSEC: IKE. Initial keys generation and periodic refreshment</i>	<i>YES</i>	<i>YES</i>	<i>HIGH</i>	<i>HIGH</i>	<i>Windows</i>
<i>IPSec</i>	<i>Machine-based (IKE). Packet data origin / integrity authentication</i>	<i>Previous (static) configuration. No tunnel management. NAT-incompatible</i>	<i>IKE. Initial keys generation and periodic refreshment</i>	<i>NO</i>	<i>NO</i>	<i>Intermediate</i>	<i>HIGH</i>	<i>Windows, Linux, FreeBSD</i>

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 [10] and [11]. This proves WEP to be insufficient to protect data flowing across WLANs.

IEEE 802.1x [6] 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. It ensures the following security properties: entity authentication and key distribution.

The 802.1x standard is intended to solve the key delivery problem of the wired equivalent protocol (WEP) with the help of 802.1x. A key management protocol is to be achieved - which provides the STAs with keys automatically. For added security, the keys can be changed rapidly at set intervals. The more often the keys are changed, the better security is achieved.

Essentially, no significant security flaws have yet been discovered in IEEE 802.1x (it is a brand new standard). However, WLAN environments using IEEE 802.1x keep relying on WEP as encryption 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 in the Windows 2000 operating system: PPTP, L2TP/IPSec and IPSec. L2TP/IPSec consists on protecting L2TP with IPSec. comparison of their properties is made 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 an entity-based authentication protocol (IKE). This could imply a weakness in multiple-user machines, in which perhaps not all the users are authorized to make use of the tunnel. However, that is not likely to happen in the WLAN: most probably, the Mobile Nodes are single-user notebooks. It also protects the packets with data integrity / origin authentication trailers, while PPTP or L2TP do not. IPsec key management scheme (IKE) is much more flexible than that of PPTP or L2TP. It allows powerful authentication options, such as x509 certificates and Kerberos.

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. 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 drawback is that all the configuration must be performed previously and manually. However, IPsec tunnels require no further maintenance. PPTP and L2TP tunnels can traverse NAT, while IPsec cannot.

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. 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 claims to protect multicast traffic, but that does not afford true multicast 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 can get to generate a lot of extra traffic.

The performance is quite a problematic feature to compare. In terms of packet overhead, L2TP/IPsec introduces the biggest overhead and PPTP introduces the smallest overhead. Additionally, PPTP and L2TP support payload compression, and L2TP also header compression (under certain circumstances). In IPsec, compression can also be performed with the Payload Compression Protocol. The improvement achieved using these options has been not tested. 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 [8][9]. It is recommended that L2TP uses some lower-level protection, such as IPsec [12][3], due to some limitations, for example the limited protection of the L2TP tunnels. Furthermore, raw L2TP is not available in the Windows 2000 operating system.

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 bandwidth-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 [2].

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 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 actual IPSec capabilities. 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 our 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 It 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 talking to 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 2.

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 Kerberos², 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 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. With these signed

²After all, no special or sophisticated software components must be required in this solution

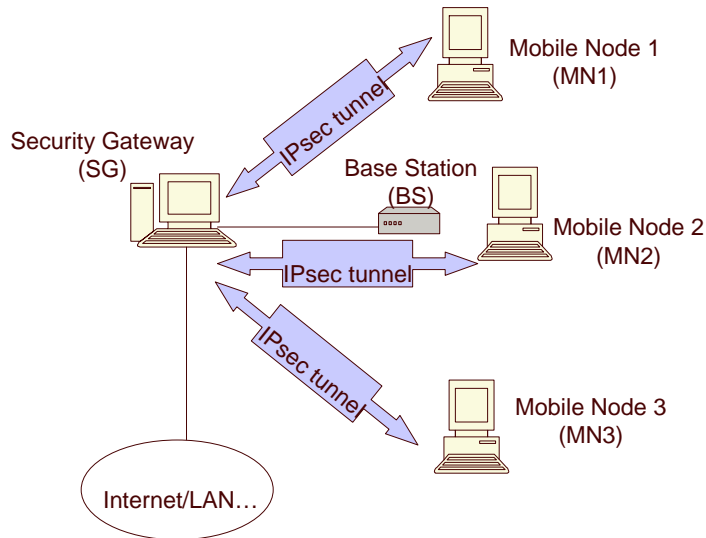


Figure 2: Scenario and Tunnel Configuration for IPsec Protection of WLAN Traffic

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 available. 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 stored under great 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 an unsafe approach.

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. These points at the need for an IPsec tunnel 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 session keys.
- Preshared secret: a pre-shared secret is present in both communicating entities (Mobile Node and Security Gateway) before the negotiation protocol takes place. From it, they can derive the same IPsec Preshared Key without sending the authentication-sensitive information (the pre-shared secret) though the wireless link. It can be seen as the Mobile Node's "password" in a certain Security Domain.

Databases 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.

Further configuration Up to now, IPsec seems to fit fairly well in the proposed scenarios' security needs. However, it has some drawbacks. The configuration of the IPsec policies is assumed to have been performed by hand before the tunnel can be established. 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 [4], based on some information generated independently of IPsec. That implies that some additional authentication must be provided somehow, in order to derive the authentication material (Preshared Key) used in the IKE authentication.

Thus, some functionality must be added to IPsec in order to facilitate an extra authentication and negotiation of the IKE's Preshared Key, as well as automate the IPsec policy configuration.

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 is the connections of all the Mobile Nodes with the Security Gateway.

Every Mobile Node establishes an IPsec tunnel with a Security Gateway, a dual homed host that acts as the router between the mobile nodes and the wired network infrastructure, as well as the IPsec association counterpart for every mobile node. This can be viewed in Figure 2. 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, who 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.³

The Mobile Nodes only accept traffic coming through 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 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 still talk to each other using the WLAN's bandwidth, since no access control is performed in the Base Station.

5 Configuration solution

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. **Manual**: it must be accomplished only once, before any other configuration takes place.
2. **Initial**: performed after the manual configuration, every time a Mobile Node needs to be registered in a new Security Domain.
3. **Automatic**: only after the manual and initial configuration have been completed; it takes place every time a user enters a new Security Domain and wants to negotiate an IPsec tunnel with the corresponding Security Gateway.

The software components involved in each configuration step are written in italics.

5.1 Manual 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 used): 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 used): 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.

³This 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.

- 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. It can be done 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 a risky 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 block (*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 used): it is useful to obtain the IP address 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 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.

5.2 Initial configuration

These configuration step takes place each time a Mobile Node needs to be registered in a new Security Domain. As pointed out in Section 4, in the entities taking part in the IPSec tunnel protocol (the Mobile Nodes and the Security Gateway) it is necessary to keep a database 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.

5.3 Automatic 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 (that is, the client's name)
 - its IP address (recently acquired for this Security Domain)
 - a random number generated by the WLANClient.

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

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

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 an error frame back. 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 (which is the Security Domain's name)
 - The Security Gateway's IP address
 - The Mobile Node's name
 - The Mobile Node's IP address
 - a random number generated by the WLANServer
 - the random number generated by the WLANClient
 - A HMAC signature of all this information.

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.

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

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.

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

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). In theory, only the Mobile Node 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 Mobile Node.

If the node’s signature is not authentic, the packet is dropped. Otherwise, the WLANServer updates its IPsec policy, adding the following rules:

- a tunnel allowing the normal traffic between the Mobile Node and any other IP address to flow, encrypted, only through the Security Gateway. This tunnel uses Preshared Key (SK) authentication.
- a rule allowing traffic which corresponds to the negotiation protocol (UDP packets to the protocol-specified ports) between the peer entities (that is, between the Mobile Node and the Security Gateway).

The Preshared Key (SK) for the IPsec tunnel authentication is dynamically produced by the WLANServer. It is the result of an HMAC feeded with the pre-shared secret between the Security Gateway and the Mobile Node (the node’s password for the present Security Domain), 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.

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 which 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.

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

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

5. Upon receipt of this packet, the WLANClient updates its IPsec settings. The settings include the following rules:
- a tunnel allowing the normal traffic between the Mobile Node and any other IP address to flow encrypted only through the Security Gateway. This rule uses Preshared Key (SK) authentication.
 - a rule allowing traffic which corresponds to the negotiation protocol (UDP packets to the protocol-specified ports) between the peer entities (that is, between the Mobile Node and the Security Gateway).

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 over the Security Gateway, by which the IPsec association between the Security Gateway and the Mobile Node is formally set up.

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 passwords⁴.

⁴For this reason, it is recommended to use a strong random password generator. More precisely, what is needed is high-entropy 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

6 Conclusion and future work

The IEEE's 802.11 Working Group is now developing a next-generation WEP but currently has no proposals for a backwards compatible encryption scheme. In fact, the only encryption scheme that is getting support from the IEEE is an AES-based proposal. A unified standards-based security framework would take, anyway, a period of time to reach the market.

VPN solutions were not originally intended to secure wireless environments. However, they have turned out to be a valid solution to the WLAN security problems. This project is an example of how VPNs can be adapted to wireless environments. It can also be considered as an analysis of the present LAN and VPN technologies landscape and the adaptability of each technology to WLANs.

IPSec has been deployed on a WLAN, securing the traffic of the legitimate users and protecting the associated wired infrastructure. For this, some software components have been added to the WLAN entities and the architecture described in Section 4 has been proposed.

The solution has, however, some limitations and future work is needed. IPSec does not protect multicast or broadcast traffic. This is an open issue that requires further analysis. Furthermore, its authentication is machine-based. In a multi-user Mobile Node, there is no way to limit the access of each user to the WLAN IPSec tunnel.

As a prototype, the software implemented in this project may be not user-friendly enough for the average users to make use of the advanced functionality of the components. GUI environments or control Applets that constitute an interface between the users and the actual software added components would be convenient.

References

- [1] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. Draft Paper, Jan 2001. <http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf>.
- [2] Microsoft Corporation. Virtual Private Networking. Windows XP Documentation, 2001. http://www.microsoft.com/windowsxp/home/using/productdoc/en/default.asp?url=/WINDOWSXP/home/using/productdoc/en/sag_IPSEctunnel.asp.
- [3] Patel et al. Securing L2TP using IPsec. RFC 3193, November 2001.
- [4] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). Internet RFC 2409, 1998.
- [5] 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.
- [6] IEEE. Standards for Local and Metropolitan Area Networks: Standard for Port Based Network Access Control. The Institute of Electrical and Electronics Engineers (IEEE), IEEE Draft P802.1X/D11, 2001, 2001.
- [7] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*, February 1997. RFC 2104.

- [8] B. Schneier and Mudge. Cryptanalysis 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.
- [9] B. Schneier, Mudge, and D. Wagner. Cryptanalysis of Microsoft's PPTP Authentication Extensions (MSCHAPv2). Counterpane Systems, 1999.
- [10] A. Shamir, I. Mantin, and S. Fluhrer. Weaknesses in the Key Scheduling Algorithm for RC4, August 2001. http://eyetap.org/~rguerra/toronto2001/rc4_ksaproc.pdf.
- [11] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the Fluhrer, Mantin and Shamir attack to break the WEP, 2001.
- [12] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer Two Tunneling Protocol "L2TP". RFC 2661, August 1999.