invent

may 2002

technical white paper

wireless security

Prepared by John Rhoton
Emerging Technologies

Hewlett-Packard Company

Reviewed by:

Jan de Clercq
Tony Pitt

abstract

Most industry analysts project mobile Internet devices to outnumber fixed devices by the end of the decade or possibly even much sooner. While the adoption rate to date has been impressive it has also met with great resistance due to security concerns.

This paper examines some of the issues related to wireless and mobile security and explores the solutions that are available to address any points of vulnerability.

contents

wireless security

Most industry analysts project mobile Internet devices to outnumber fixed devices by the end of the decade or possibly even much sooner. While the adoption rate to date has been impressive it has also met with great resistance due to security concerns.

This paper examines some of the issues related to wireless and mobile security and explores the solutions that are available to address any points of vulnerability.

In many ways wireless security is just like wireline security. The issues are largely the same. Regardless of the medium every system needs to safeguard proper authentication, privacy of transmission, prevention of viruses and protection against denial-of-service attacks.

The differences arise from the fact that mobile devices and transmission over an unshielded medium (air) are inherently more vulnerable to impersonation, sabotage and interception.

device security

problems with mobile passwords

Keeping passwords secret is a challenge in any environment. But the nature of mobile devices makes them even more susceptible than fixed terminals. The miniaturization of the computer, which implies small user interfaces and keypads, leads users to select even simpler passwords than they would in the wired world. Since multiple letters are associated with each numeric key on a keypad (differentiated through iterative key-presses: e.g. press once for A, press twice for B) many users choose words that use the first (single-press) letters, thereby substantially reducing the number of possible passwords.

Many phones have "intelligent" entry, such as iTAP or T9, which will perform dictionary comparisons to make text entry using a phone keypad easier. The user needs to enter fewer key-presses per letter. If this software is activated during password entry, the user may be more inclined to choose longer passwords, however the combinatorial possibilities are still reduced since the keypad uses less keys than a keyboard.

Mobile devices are used more frequently in public, making the chance of shoulder-surfing more real. While you have some control over who stands behind you in your own office it is virtually impossible to prevent someone from standing behind you (often without your knowledge) in public places, like on airplanes, in lounges and trains.

Many users cache their passwords on their device in order to automate their connection to a server. Stolen and lost machines then provide pre-configured access to the corporate network and applications.

Mobile phones and some PDA login programs can provide an intermediate solution to the mobile password mechanism. They provide limited attempts (between one and three, typically) to log in with a simple PIN, after which a more complex password or longer PIN is required. This provides a compromise, such that for normal access the user can log in simply, but the hacker cannot readily gain access through a brute-force attack.

vulnerable file system

Misplaced and misappropriated devices also represent another security threat. If they contain sensitive data (such as passwords and account numbers or classified and/or confidential information) it is usually not difficult to obtain it once the device changes hands.

Even password-protected devices often have some means of bypassing security, just in case the user forgets the password. For similar reasons, they usually do not lock out the user when confronted with a brute-force attack.

It is important to verify what happens to the data and security if the device is powered down and/or completely reset. In many cases it is possible to physically extract the data if the motivation is high enough to justify the effort.

While desktop operating systems, such as Windows 2000, offer an encrypted file system, this is not yet common on mobile platforms. If the data is sensitive it is therefore worthwhile to investigate additional security software, such as F-secure, which can address this issue. These products combine the encryption of all or part of the file system with enhanced login security.

Whilst the file system on the device itself is somewhat protected by the login security provided, data on external devices is not similarly protected. The density of information storage is a problem here – the ability to store 1Gbyte of information on a 1 inch square Microdrive at a cost of a few hundred dollars will encourage people to carry more and more confidential information when they travel. However, it is very easy to mislay something this small, and unless the data is encrypted on the device it can then be read on another device.

Encryption of data on external media (both compactflash and Microdrive) should be automatic, and tied to the user's login information. That way, if the media is lost, the data does not fall into unauthorized hands. If the encryption is not automatic, then users will find it easier not to bother using it.

smart cards

A partial solution to the problems of mobile devices is the use of Smart cards. These hardware tokens provide secure storage of PINs/Passwords, private keys and certificates. They also provide a tamper-proof implementation of the cryptographic algorithms (which are usually based on elliptic curve cryptosystems and therefore less processor memory intensive than the otherwise more common RSA algorithms).

Obviously these could just as easily be stored on the device itself. What makes these unique is that the tokens are virtually tamper-proof and can only be used after authentication to the card.

That is not to say that it is theoretically impossible to crack a smart card. Some of the attempts have included modifying the circuits, erasing EEPROM [Electrically Erasable Programmable Read Only Memory] (for example, by manipulating voltage levels), and attacking the Random Number Generator, which influences the generation of the cryptographic keys.

What is important is that all these attacks operate at a physical level. That is to say that they require physical access, sophisticated tools and advanced expertise, which include a comprehensive understanding of the physical layout of the card.

smart card standards

There are several different types of standards related to smart cards. In particular, there is the physical interface to the card. While this interface would conceivably suffice it is not practical for applications to operate at this level. In the same way that networks are divided into layers for ease of use, you can imagine the application-programming interface to be positioned above the physical interface.

Most of the physical interfaces to IC [Integrated Circuit] cards are developed collaboratively by ISO [International Organization for Standardization] and IEC [International Electrotechnical Commission]. The most common of the physical interfaces is ISO/IEC 7816. It defines the card size; placement, shape and size of the contacts; the purpose of each of the contacts; the voltages and signals that are applied to the clock and I/O ports; the command set recognized by smart cards. In a nutshell it covers the whole protocol that goes over these contacts.

The other physical interfaces do not assume physical contact but instead use optical interfaces (ISO/IEC 11694) or wireless (radio-frequency) interfaces (ISO/IEC 10536, 14443, 15693) covering varying ranges.

PKCS [Public Key Cryptography Standard] #11, also known as Cryptoki [Cryptographic Token Interface] is the de facto standard and most common API to cryptographic function. It was developed by RSA Security and is available across most platforms.

There are two other common application-programming interfaces to smart cards. The PC/SC [Personal Computer / Smart Card] interface was developed by a group including HP, IBM, Sun and Schlumberger, as well as several others. Microsoft drove the specification and continues to implement it on most Windows platforms. It has the advantage of being more functional that PKCS #11 but is not as portable.

The OCF [OpenCard Framework] is a more Java-oriented approach with many similarities to PC/SC. And that includes both the advantages and disadvantages.

It is important to distinguish both OCF and PC/SC from JavaCard and Windows for Smartcards. The latter two are operating systems that run on a card, whereas the former run on a computer and serve to abstract the (smart card) communication details from the applications.

RSA also has another related standards: PKCS #15, the Cryptographic Token Information Syntax Standard. This syntax defines the format of cryptographic tokens (e.g. secret or private keys, authentication objects), which are stored on an IC card. This is regardless of the cryptographic interface used to transfer them.

SIM cards

SIMs [Subscriber Identity Modules] are also forms of smart cards. They have become popular with the GSM system where they are a core part of the specification. The SIM cards authenticate a user to a mobile operator and provide functions for securing the transmission.

GSM SIM cards also use the ISO/IEC 7816 standard. However, they do not use a “standard” application-programming interface since there was no intention for the SIM cards to be used on a grand scale by application programmers. There is a SIM Toolkit, which can be used to exploit the additional capacity of the cards.

Cards with SIM Toolkit applications can monitor the phone’s keyboard and dynamically insert menus. These menu items then trigger applications, which can request information from the end-user (e.g. a credit card account number or dollar amount).

As shown in the table below, GSM was the first but will not be the only mobile phone system to use smart cards.

- CDMA (IS-95) R-UIMs have been tested in Asia and will be appearing on the market soon.
- UMTS has already specified the use of the USIM (universal SIM) as the evolution of the SIM.
- ▪ The GSM ANSI-136 Interoperability Team (GAIT) has finalized the specifications of a multi-mode phone supporting GSM 900/1800/1900, TDMA (IS-136) 800/1900, and AMPS. The GAIT phone also relies on the SIM card for the storage of all GSM and IS-136 authentication, roaming, and service information.

Abbreviation	Full name	Defining Standard
SIM	Subscriber Identity Module	GSM
WIM	Wireless Identity Module	GSM / WAP
UIM	User Identity Module	IS-95
R-UIM	Removable UIM	IS-95
USIM	Universal SIM	UMTS

Smartcard based phone identity modules

Note that the SIM cards offer additional advantages to the user besides enhanced security. They decouple the handset from the mobile subscription, which facilitates global roaming, as there is no longer the requirement to use the same air interface. According to projections from the Gartner Group (Mobile Value Added Services and Smart Cards, July 2000) more than 80 percent of all mobile terminals shipped will contain a smart card (SIM, R-UIM or UIM) by 2004.

virus protection

Not all mobile configurations are vulnerable to virus attacks. In order for a virus to install itself it must first be transferred to the device. This implies connectivity. Then it must execute on the device, which implies the ability to run ad-hoc applications. The damage a virus can do depends on the privileges and functions available to these applications. Many can modify local storage and therefore perpetuate themselves on the device. In order to be effectively distributed they need to have some means of propagating themselves to other users. Sending an e-mail message to all the recipients in the address book is the usual approach but other mechanisms (posting files for HTTP/FTP download) are also possible.

For a sophisticated and open system like a desktop it is easy to meet those requirements. Laptops and PDAs are already exposed to the same risk whether they use wireless or fixed connections.

But most mobile tools are still very limited and therefore less susceptible. However, it would be naïve to assume that they are likely to remain immune for the foreseeable future. As they grow in power and open their systems they too will be the target of hostile attacks.

WAP phones have not yet encountered significant problems, mainly because most WML is passive. However WMLscript opens the door to active applications. In particular, the telephony interface, WTAI, is a potential area for misuse, which could conceivably include expensive phone charges incurred without the knowledge of the user.

The fundamental problem with virus protection is how to defend the system without restricting constructive applications and data. It would be simple to close all the doors but that would also reduce the usefulness of the machine.

The most common approach is to scan through any executable code looking for a known set of viruses. The biggest question is where to run this search. Enterprises frequently scan at the firewall or on mail servers. But mobile devices are often connected to the Internet without using the corporate network. So it is difficult to ensure that they are not infected.

The solution for laptops may be to run some virus protection software locally. Smaller devices, however, often lack the computational resources to be able to do this efficiently although there has been some progress at least on the PDAs. F-secure, for example, has an anti-virus solution for the iPAQ Pocket PC.

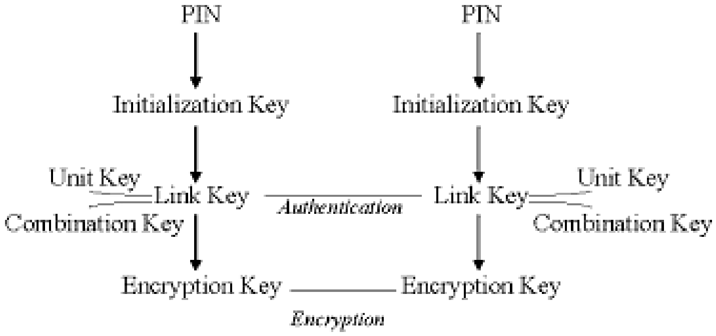
Even local solutions have a problem with mobility though. New viruses appear regularly and only a machine that has the properly updated virus signatures is optimally protected. It is therefore necessary to ensure that the newest signature files from the Anti-virus vendor are redistributed to all devices frequently.

air security

Bluetooth

Bluetooth is intended for "Personal Area Networks", as a means of connectivity between personal devices and peripherals. The short range of Bluetooth functions as a means of protection against large-scale anonymous attacks. While that might suffice for the home and other controlled environments it is clearly inadequate for public environments, where competitors and indiscriminate hackers might also be present.

Bluetooth authentication graphic



The ad hoc nature of Bluetooth prevents any security mechanism that relies on additional infrastructure. This excludes its integration within a PKI and or its use of Kerberos, which requires a Key Distribution Centre. Its authentication scheme is confined to the security that can be provided in a peer-to-peer model and can be incorporated into miniature devices.

In order to address the need for additional security Bluetooth devices can require authentication before establishing a link. A link key is then established between the devices and all communication can be encrypted.

The initial authentication makes use of a PIN, which must be identical on any paired devices in order to set up the link key. Ideally all PINs are configurable on the device but Bluetooth is also available on peripherals that do not have a UI and must therefore utilize a hard-coded PIN.

Bluetooth security includes the notion of *Authorization*. Each device can be configured with a set of *Trusted* devices, which have unrestricted access to all services provided. *Devices not trusted*, on the other hand, have access only to services that have not been restricted

It is beyond the scope of this paper to analyse the Bluetooth cryptographic algorithms. However, for the sake of familiarisation some of the more common functions are listed below:

- E0 – Encryption
- E1 – Authentication
- E21 – Unit key generation
- E22 – Initialisation key generation
- E3 – Encryption key generation

Two potential weaknesses of Bluetooth security include the PIN attack and the location attack.

PIN attack

Whether the PIN is hard-coded or configurable it is important to realize that short (e.g. 4-digit) PINs have only a small set of possible values. We compound this problem with the fact that many users tend to leave settings with their default values. Consequently the chances of breaking a PIN, for example through a brute force attack, can be significant in a poorly configured environment.

traffic analysis (a.k.a. location attack)

It is possible to recognize a device without authenticating to it. This is not ordinarily a big risk. However, it is feasible that Bluetooth probes could be installed in prominent locations, which might include airports, entrances to corporate office buildings, or even private homes of selected targets. By correlating the traffic it is then possible to identify patterns of movement that may provide insight into business dealings and other discretionary information.

WLAN

Wireless LANs are most often used in corporate environments where all employees are presumed to have unrestricted access to the network. However, close proximity is not a sufficient factor for authorization since there may be guests or neighbouring offices that share the same air space but should not be allowed to access network resources.

The only way to restrict this today is through the use of Wired Equivalent Privacy (WEP), which can be configured at the access point. Another network authentication scheme, called 802.1x, is emerging that would augment WEP, if applied to wireless LANs. Major manufacturers such as Cisco, 3Com and Enterasys already have products supporting the new standard.

WEP

WEP uses a symmetrical algorithm called RC4 with either a 40-bit or 128-bit key. When WEP is enabled, each station (clients and access points) has up to four keys. It is able to provide both authentication and encryption of all data transmitted over the air. Although the algorithm will permit different keys for each user few implementations are able to manage per-user keys, and those that do use proprietary protocols that cause interoperability problems. Instead all devices using a single access point typically share one key.

It should be noted that WEP is only available when an Access Point is in use. Ad hoc networks are trivial to set up but its stations cannot enforce authentication and must pass all data in the clear.

There are two problems with WEP. The first is the cryptological weakness in the algorithm that makes it susceptible to statistical analysis. While there have not yet been reports of any actual intrusions staged on a purely cryptological basis they may be imminent, as they do not require substantial equipment.

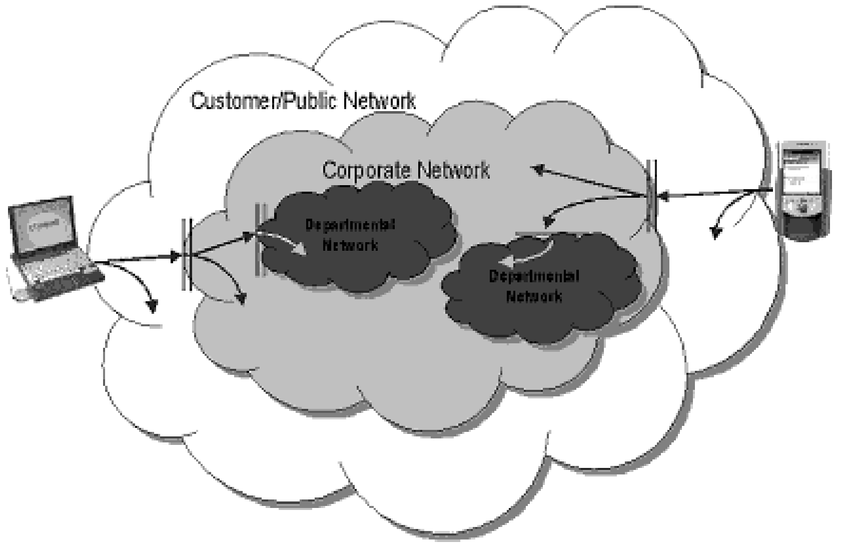
Another problem is key distribution. Since all users of one access point must share the same password, and most corporations would like hand-over between access points to be transparent to the user, it implies that all users must use the password. This poses a problem if an employee resigns or is terminated. Theoretically, the password should then be changed but it is not a simple task to communicate the new password to the entire user-base in a timely manner. The larger and more dynamic the employee population the more complicated the process becomes.

As a consequence of these issues the next version of 802.11 (802.11a) should include security enhancements that address the weaknesses. One part of this is 802.1x, which can be included in any access point and will permit authentication to any authentication database. Per-user authentication eliminates the key-distribution problem.

802.1x

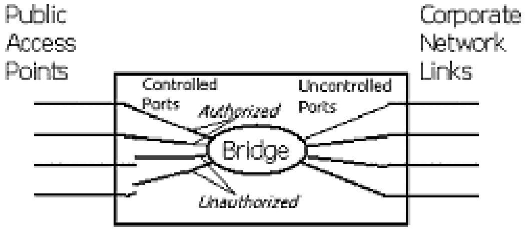
802.1x is not limited to wireless networks. It can be used to authenticate user access to any closed network. For example, a company may have a private network, which should be accessible only to employees, with some more public segments, that can also be made available to customers. Without 802.1x it would be necessary to isolate these two networks, which could lead to significant duplication of effort and equipment.

The fundamental approach used by 802.1x is to authenticate users at the edge of the private network. It would be conceivable to perform this processing at other points within the core of the network, for example using MAC addresses. However it would be difficult to protect all authenticated end stations from unauthenticated stations, since intruders could bypass authentication at least on their own segments. It is significantly less complex, and more scaleable, to ensure security if the authentication is performed on the external boundary of the network.



It is possible to develop a tiered authentication scheme in which the public is able to access the external network. All employees can access the corporate network, and individuals can access their restricted departmental LANs.

controlled ports



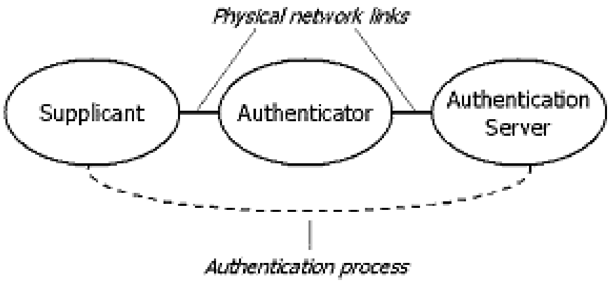
A typical bridge would connect segments that are private and presumed to already be secure, such as those on the corporate network. An 802.1x bridge can connect these segments too but its added value lies in its ability to optionally authenticate a port before allowing it to connect.

What this means is that the bridge is configured with both controlled and uncontrolled ports. Those that are uncontrolled do not need to authenticate and would see the device as though it were a traditional bridge. Devices connecting to the controlled ports would not be able to access any of the connected segments (neither the segments on the uncontrolled ports nor the segments on authenticated controlled ports) until they authenticated successfully.

802.1x architecture

The scenario sounds simple in principle. Where it becomes slightly more complicated is in the actual authentication. Conceptually it would be feasible to let the bridge perform the authentication using a cache of authentication information. However, that would be unnecessary overhead for the bridge and would mean that authentication information would need to be replicated to all bridges, which is neither efficient nor secure.

Instead the bridge (called the Authenticator) may relay authentication requests from a client (called the Supplicant) to an Authentication server. This is very similar to the RADIUS model of authentication, and, in fact, it is expected that many Authenticators will be RADIUS clients -- and many Authentication servers therefore RADIUS servers.



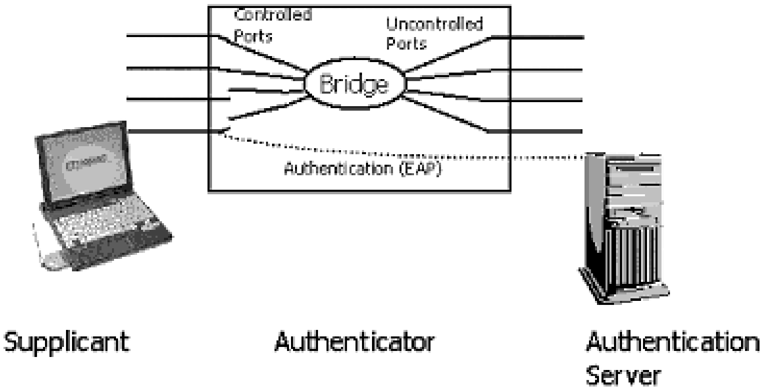
There are three players in this topology. The Authenticator sits in the middle with both controlled and uncontrolled ports. The Authentication server is connected to an uncontrolled port. The Supplicant is connected to a controlled port.

The authentication process would run along these lines:

1. The Supplicant connects to a controlled port
2. Either the Authenticator or the Supplicant initiates authentication
3. A challenge is sent from the Authentication server to the Supplicant via the Authenticator
4. The Supplicant signs, or otherwise cryptologically processes, the challenge.
5. The Supplicant sends the result back to the Authentication server via the Authenticator
6. The Authentication server sends the status (success or failure) back to the Supplicant via the Authenticator
7. The Authenticator intercepts the status and, if successful, opens the port.

There are a few issues to consider in this process:

- The only traffic that the Authenticator may relay from/to a controlled port is authentication requests/responses
- For security reasons, the authentication information must be cryptologically secure. This implies that the Authenticator cannot decrypt the credentials.
- The model must be extensible to new authentication mechanisms as they are invented and implemented.



To ensure that the Authenticator can always identify and interpret new authentication mechanisms, any authentication types must be encapsulated using the Extensible Authentication Protocol (EAP) as specified in RFC 2284. EAP already supports multiple authentication schemes including smart cards, Kerberos, Public Key Encryptions, One Time Passwords. And many others can be added.

security considerations

The biggest security consideration of 802.1x is that its sole purpose is authentication. It does not provide integrity, encryption, replay protection or non-repudiation. These would need to be implemented with complementary schemes such as IPsec.

There are also other points of vulnerability that must be addressed in any implementation of 802.1x.

- Piggybacking on an authenticated port – Multiple end stations on a port must be detected and disconnected
- Interception of credentials – Passwords must always be encrypted
- Subversion of authentication negotiation – It should not be possible to provoke a lesser form of authentication by interfering with the authentication process

wireless implementation

802.11b Wireless LANs are ideal candidates for 802.1x authentication since they represent a completely uncontrolled periphery. While it is possible to restrict physical access to wired LANs this is not feasible in a wireless environment. It is much more difficult to monitor and enforce the air space around office buildings than the ports and wiring within them.

This vulnerability is currently addressed using Wired Equivalent Privacy (WEP), which is available on 802.11b Access Points. If in use, then all stations must configure a symmetric passphrase in order to connect. All transmission is then encrypted with 40-128 bit encryption.

Recently, there have been alleged cryptological weaknesses with the WEP algorithms that have cast a shadow on its use. Beyond these there is a fundamental problem with key distribution and update. Since WEP keys are typically symmetrical (the same on the Access Point and all connecting stations) they need to be changed in unison. Clearly this is difficult to orchestrate when large user populations are involved.

There have been solutions, including automating regular key changes, for example, using logon scripts, however, they are non-standard and require additional work. There are also problems ensuring that employees who leave the company no longer have access to the network, since they could “remember” their WEP key.

Another aspect of the problem arises when users connect to multiple different wireless LANs (e.g. in public areas or at customer sites). Current WEP implementations require that the user manually change the WEP key each time a new network is selected which is tedious and interferes with any automated key changes.

802.1x solves all of these problems. It is not necessary to distribute any keys. The user can authenticate to a central Authentication server, which stores per-user credentials that can be disabled or modified as needed.

This does not mean that there is no longer a need for WEP in an 802.11b LAN. As mentioned above, 802.1x only provides authentication. It does not encrypt the over-the-air transmission. It is therefore still possible for hackers to eavesdrop on conversations and intercept sensitive information.

The ideal combination is to use 802.1x for authentication to the network but WEP to ensure privacy of the transmission. This does not address the cryptological weaknesses of WEP, however it does open the door for future versions of WEP to focus on privacy rather than authentication.

WWAN

Digitized mobile telephone and wireless packet data networks all include some kind of encryption. First generation (analog) systems are not suitable for data transmission so the risk of intrusion was limited to eavesdropping on private conversations and no encryption was deemed necessary. Since the migration to the second generation of wireless telephony is virtual complete we can restrict this discussion to digital networks.

The world's most widely used wireless phone system is GSM, which uses a smart card that contains both the IMSI (International Mobile Subscriber Identity) and subscriber identification key. Upon establishing a connection with a mobile base station a session key is negotiated and all transmissions, voice and data, are encrypted and very difficult to crack.

GSM documents specify the rough functional characteristics of its protocols including the secure encryption of transmitted digital messages. However, apart from the protocols, details of the algorithms are kept secret. Most security specialists will argue that security by obscurity is not an effective approach, since only the close scrutiny of a large set of experts can ensure that there are no obvious weaknesses in the technique. Nonetheless, GSM contains 3 secret algorithms, which are only given to vendors with established need-to-know, such as carriers and handset manufacturers.

- A3: Authentication algorithm
- A8: Cipher Key Generator (essentially a 1-way function), and (session key generation)
- A5: Ciphering/Deciphering algorithm (presently A5/1,A5/2). Provides over-the-air voice privacy

The SIM card contains A3, A5 and A8; the base station is equipped with A5 encryption, and is connected with an authentication centre using A3 and A8 algorithms to authenticate the mobile participant and generate a session key.

Most of the other wireless wide area networks (e.g. IS-95 CDMA, IS-136 TDMA) were developed in the United States and use the CMEA encryption standard specified by the Telecommunications Industry Association (TIA), also an effective encryption technique. Additionally IS-95 CDMA uses a transmission technique called spread spectrum that was developed by the military with the express intent of making interception more difficult.

Although these encryption algorithms provide an effective barrier against the vast majority of hackers it is important to realize that they are not uncrackable. Both the CMEA and GSM Algorithms are reported to have been cracked. The value of the protection does not lie in providing a completely secure environment for very sensitive transaction. Instead it offers an obstacle so that monitoring and interception of random or bulk transmissions is simply not cost-effective.

network security

In addition to the security of the air interface of a wireless WAN we also need to consider the network between the base station and the application server. Fundamentally there are two different means a mobile network may offer to transfer data. It can provide a packet-data network or else it can use circuit-switched connections.

A packet data network is simpler. CDPD, Mobitex and GPRS would all be examples of packet data networks. In these cases the mobile device has an IP address and it transfers data through the mobile network, which is connected, to the Internet. No special configuration is typically required at the mobile end. Its data access is transparent. If the IP address given to the device is fixed then a minimal amount of authentication is also implicit in any packets originating from it.

Data communication over primarily voice networks, such as GSM, IS-136 and IS-95, is not quite as straightforward. Typically a PPP connection must first be made from the device to a dial-in server. The dial-in server will assign an IP address and relay all the traffic between the device and any application servers.

This implies some configuration at the mobile end. The phone number must be specified and then the user must authenticate to the dial-in server using an authentication protocol such as PAP, CHAP or MS-CHAP. So the dial-in server knows who the user is but the application server does not. It cannot determine the phone number easily and the IP address is meaningless. If necessary it would then re-authenticate the user, which means additional work for the user.

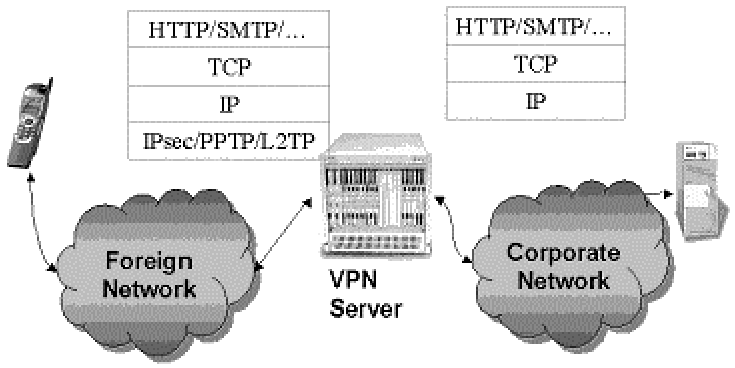
It would be possible to bypass the first authentication by storing the mobile phone number on the dial-in server and then comparing the caller-id of incoming calls. However, this would provide unlimited access to the corporate network when a device was lost or stolen.

Solutions to address this dilemma must combine security with ease of use, for example by using biometric authentication. (In 1999 Siemens showed prototypes of a mobile phone that incorporated a fingertip biometric sensor. Although not available in production at this time, such combinations of technology are clearly possible, and offer considerable advantages in the battle against theft and fraud.) They must also ensure that unauthenticated users cannot access any information on the device, for example by encrypting the file system. It is then possible to cache some of the network credentials on the device. Ultimately, however, some authentication to the network should always be based on an action or token that is separate from the device.

supplementary security

It is possible to augment the security of the air interface by creating a secure path beyond the mobile network either to a specific end-point or to the perimeter of a corporate network. In a sense, this always implies a tunnel, also known as a virtual private network (VPN). VPNs were not designed with wireless networks in mind and are therefore more prone to failure due to unreliability and low bandwidth. While the impact of this will be reduced as networks improve, it is a factor that must be considered in any current deployment.

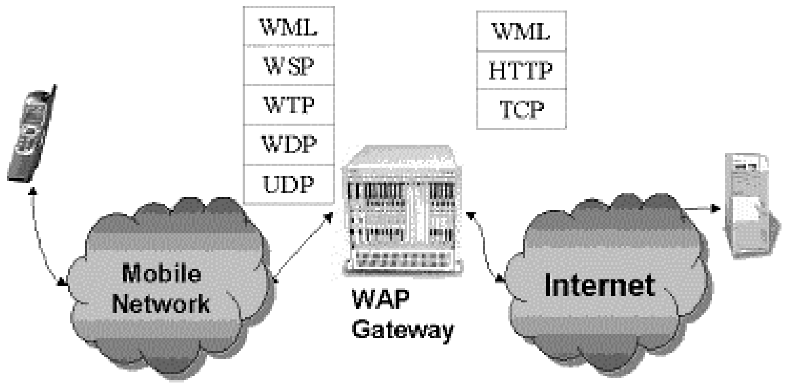
VPN diagram



There are several different VPN protocols available. Some of the more common ones include PPTP, L2TP, L2F and IPSec. But obviously both the mobile device and the server must support a common protocol. The limitation is usually found on the device, since it will often not have any VPN client available, and if it does, it is likely to be restricted in terms of which protocols it support. It isn't difficult to get tunnel servers for any of the above-mentioned standards.

The principal reason for the lack of VPN client support on device lies with the fact that conventional tunnelling protocols are not designed for small, wireless devices. IPSec for example requires too much CPU power for today's mobile phones (though this will change over time), and works particularly badly in networks with significant packet loss.

WAP diagram



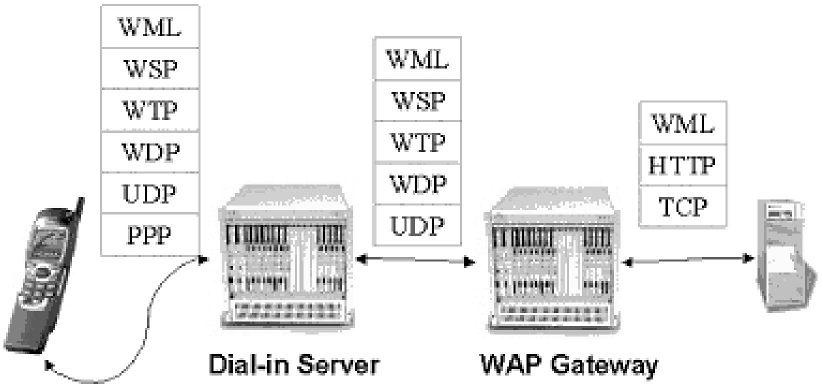
One specific VPN, very particular to the mobile WAP environment is WTLS. Wireless Transport Layer Security is based on TLS, the successor of SSL. However, unlike TLS/SSL which are intended for end-to-end encryption between the client and the application server, WTLS is contained in the WAP stack and only encrypts the path from the client to the WAP gateway. The WAP gateway will often re-encrypt the data to the application server using SSL, but the fact that it is decrypted in the WAP gateway implies that there is no end-to-end encryption. WTLS is therefore functionally more similar to the VPN protocols than to TLS/SSL.

Unfortunately, the principal weakness of TLS/SSL comes in the form of the “man in the middle” attack, where a device intercepts the key exchange, acting as server to the client, and as client to the server. By doing this, it has access to encryption keys in use for the secure session, and hence to all the data within that session. Since the WTLS session usually terminates on the WAP Gateway, with communication from there to the Web Server going via SSL, the WAP Gateway is itself exactly the “man in the middle” for that attack. The trust of WTLS is therefore tightly bound up with the trust of the WAP Gateway itself.

In addition to privacy and integrity, which WTLS always provides, it is possible to stipulate authentication requirements. All WTLS sessions are categorized into one of three classes. Class 1 is anonymous, meaning that neither party is authenticated. Class 2 implies server authentication only. Class 3 requires both the client and server to authenticate themselves by providing a signed certificate.

WAP authentication works very much like the network authentication mentioned earlier. Depending on the bearer network we may have PPP over circuit-switched data or we may simply have a packet data network (such as GPRS). In terms of client authentication this means possibly PPP authentication as well as possibly WTLS (in the case of Class 3) authentication or maximally 2 user authentications. Or, it can mean no authentication at all, for example, in the case of GPRS with WTLS Class 1.

There are, however, some additional considerations.



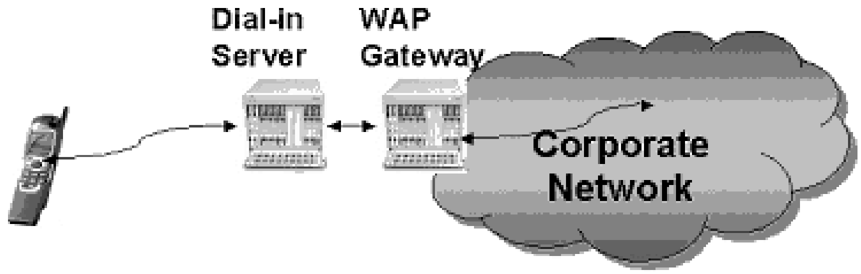
WAP gateways may also be configured to provide an additional level of authentication. This would typically work with the WAP gateway sending the user a WML form that asked for a username and password. This transaction could optionally be encrypted using WTLS so that the WAP Gateway credentials could not be intercepted.

That is an additional feature of WAP authentication which gets around some of the drawbacks associated with the WAP devices currently available.

One big drawback is that (since WAP devices are not an open platform) it is not possible to influence the authentication procedures or add additional ones. In particular some devices do not support all the common authentication protocols. For example, Nokia phones do not work with MS-CHAP. And most of the phones with Phone.com (OpenWave) browsers do not allow backslashes, which makes multi-domain authentication difficult for Microsoft shops.

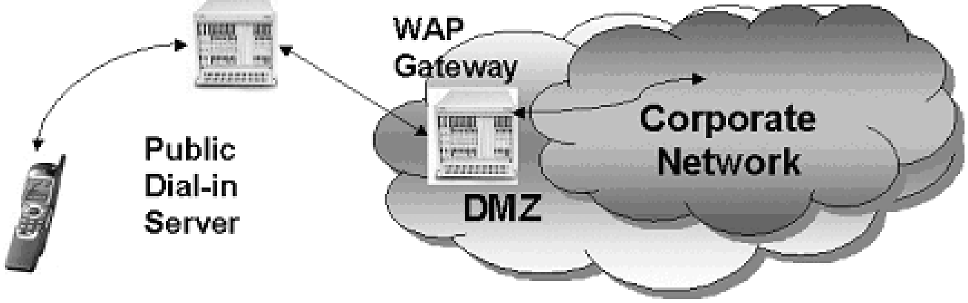
A further risk to security is the fact that WAP phones often store the credentials on the phone. This opens up the possibility that someone with a stolen phone could extract the credentials and attain unlimited access to the corporate network.

There are two ways to get around this. Either you set up a dedicated WAP dial-in service which uses trivial credentials and instead relies on the WAP gateway to perform the authentication. Or else you rely on public PPP servers (i.e. ISPs) to provide the dial-in service and place the WAP gateway in a DMZ where it acts as a tunnel server to provide access to the corporate network.



In the case of dedicated dial-in service you would need to ensure that the PPP service and the WAP gateway are either collocated on the same machine (with no IP forwarding) or else use a dedicated network link. The dial-in service would then only provide access from the WAP device to the WAP gateway. After authentication the WAP gateway would effectively function as a proxy and would relay all the WAP traffic (converted to HTTP) into the corporate network and vice versa.

With this scenario the credentials of the dial-in service would not be sensitive since they would not be used for any other purpose. The "real" authentication would occur at the gateway and, even if the PPP credentials were jeopardized they would not pose a threat to the corporate network.



Another approach would be to allow users to dial into an ISP and connect from there to the corporate WAP gateway. By placing a WAP gateway in the DMZ it could be protected against many attacks (by only allowing WAP traffic to pass through the external firewall to it). It could also perform authentication of the user. And the internal firewall would be configured to only allow HTTP/SSL traffic from the WAP gateway.

Note that both WAP configurations described above assume that the device is configurable. Many mobile operators (particularly in North America) sell WAP phones with hard-coded configuration settings. The first approach requires that both the phone number of the dial-in server and the IP address of the WAP gateway be configurable. The second scenario requires only a configurable WAP gateway.

If these conditions are not met then the only recourse is to rely on the security of the mobile operator. This paper recommends against any such approach since it is not possible to verify or enforce any level of security.

application connectors

An alternative approach to network or session level tunnelling is the use of Application-specific connectors, such as those available from Infowave. Client software on the PDA communicates in a secure manner with a server in the DMZ. This server then communicates with the application server on the Corporate Network. Existing products are designed to work today's wireless networks, coping well with the limited and variable bandwidth available, and recovering gracefully and efficiently from packet loss. These techniques could be extended in the future to implant the client software on mobile phones.

The particular benefit of the application connector approach is that the software can be tailored very specifically to the application it is supporting, thereby providing the best possible communication over whatever network is available.

enterprise requirements

Enterprises typically have a dual-level security structure. The first level is the perimeter of the corporate network. In order to reduce the threat of industrial espionage or deliberate sabotage, only employees and authorized contractors are allowed any access into the network. While this safety net is difficult to enforce 100% it does thwart the attempts of casual hackers and create an additional obstacle for sophisticated intruders.

Beyond the common perimeter a second level of security may protect the data and applications on an individual basis.

perimeter security

What does this mean for wireless implementations? Firstly, the secured perimeter must be accessible to mobile devices. Secondly, access to the perimeter from the mobile device must be encrypted, in order to ensure that it is not intercepted or falsified. Typically, the solution to both of these means the use of a VPN. It is not simple, however, to find mobile devices that support VPNs at this time.

As we try to strengthen client authentication there are two directions we can look. The first is multi-credential authentication. We can require a password (PIN) to access the device. Then we can require another set of credentials to access the corporate network. And finally we can request further authentication from each sensitive application. While cumbersome to the user this approach does permit a tiered authentication scheme, which will reduce the impact of any compromised credentials.

Another dimension of authentication is multifactor authentication. In addition to merely entering a memorized username and password ("something you know") we can require other forms of authentication. This can range from various types of removable and/or contactless smart cards ("something you have") to biometric techniques including fingerprint scanners and voice recognition ("something you are"). A combination of several of these techniques can provide a very effective security scheme.

The most sensitive applications need to maintain an additional level of security configurations that include Authentication, Authorization and Auditing. Users who have a business need to access the application must authenticate to the application before they can use it. Depending on their role and responsibilities they may be given different authorization levels (e.g. read-only, modify, delete) or authorization only to certain subsets of the data. All the actions requested and performed are logged to preserve an audit trail.

Most of this security is independent of the means of access. If it works for fixed lines then it should also work for wireless users. The only additional factors that need to be considered in a wireless context are whether the user will require two sets of credentials (for mobile and fixed usage), and whether some kind of single-sign on will be supported. Particularly given the cumbersome entry of passwords on some mobile devices, it is not user-friendly to require them to be entered separately for each application. At the same time, mobile credentials are more likely to be compromised so the risk must also be minimised.

secure transactions

Excluding military applications the most sensitive types of transactions are financial. Once money is involved, the incentive is clearly there for criminals to attempt to intercept and/or falsify any transmission. Depending on the value of the transaction the incentive may be quite substantial and can cost-justify an expensive attack on a financial system.

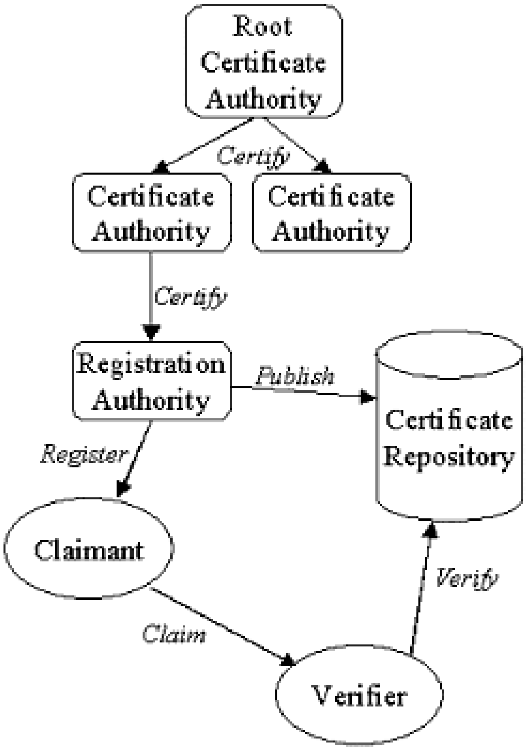
The main differences between transactional security and enterprise security, described above, are:

- The absence of a perimeter
- The absolute requirement of integral end-to-end security, including privacy, integrity and, above all, non-repudiation

A successful implementation of transactional security can serve as a platform to enable any type of wireless financial transactions, including mobile banking, mobile stock trading, mobile commerce, mobile betting/gambling, as well as other non-financial but still legally binding transactions such as reservations.

public key infrastructure

Most approaches to achieve this security involve public key cryptography. This does not simply mean installing of a piece of software on the device and the application server. A full public-key infrastructure must be implemented including certificate directories, certification and registration authorities as well as adequate revocation checking protocols. These may be mobile specific or general purpose PKIs.



Typically one or more Certificate Authorities (CAs) are chained together in a hierarchy. This allows Verifiers to only be seeded with a minimal number of Root CAs. They can then verify the validity of the actual CA by validating each certificate in the chain.

In addition to the CAs, which primarily produce certificates (although they can also publish them), it is also necessary to register certification candidates. This means verifying their identity, retrieving a certificate from the CA, publishing that certificate in a globally visible repository and also passing it on to the claimaint.

The Claimant eventually makes use of his/her certificate by passing it to the Verifier. The Verifier can then validate the certificate. This means processing the internal structure of the certificate to ensure consistency, checking that at least one of the CAs in the chain is trusted, and finally also checking the (revocation) status of the certificate, for example in the Certificate Repository.

This is a simplistic view of a typical PKI. It entails a number of challenges that must be addressed. The RA must follow careful procedures at registration to ensure that a certificate is not issued to an impersonator. But even if it is correctly issued there is a risk that the user could abuse it or that it could be compromised.

In some of these unplanned circumstances it is important to define the consequences and procedures that must be followed. Certificate revocation is a complex aspect of a PKI but one that is very important in order to limit the financial and legal liability of each of the entities.

If we focus in on the notion of a wireless PKI there are several services that can be identified. Server certificates are needed for all the application and infrastructure servers involved in the configuration. Clients will also require certificates if they need to authenticate themselves. The whole certification infrastructure may be independent or joined with a wired PKI but, one way or the other, the clients need to be configured to trust the root CA.

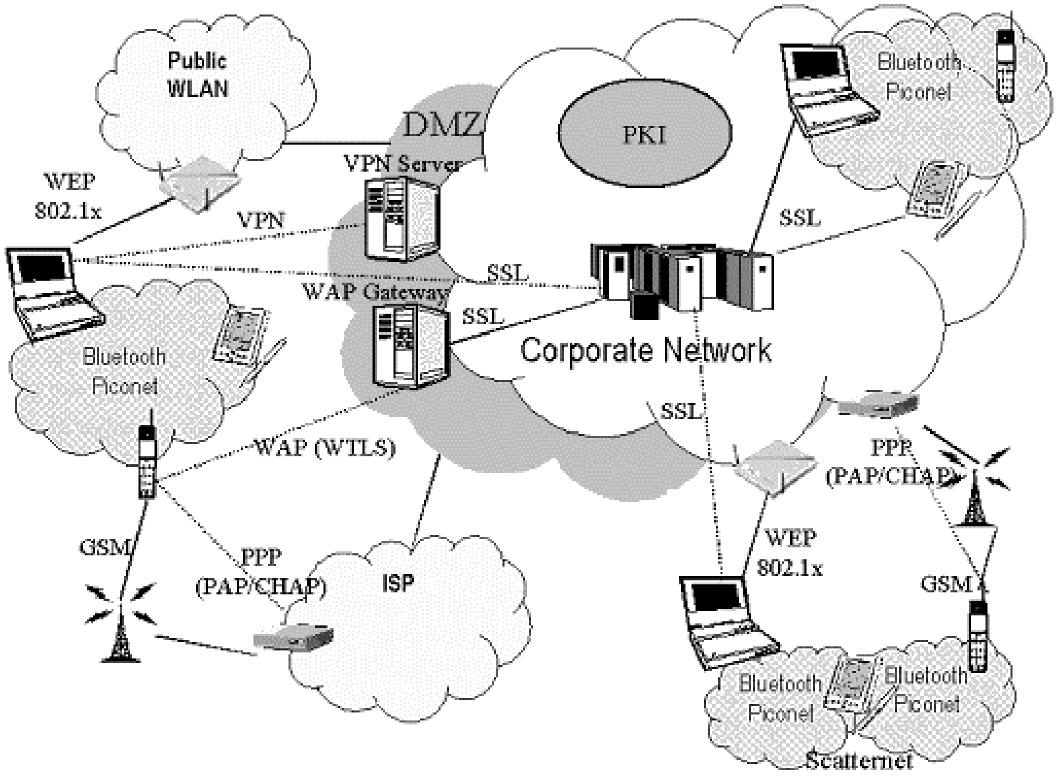
Further requirements might be the validation of certificates, and in particular, the publishing and accessibility of Certificate Revocation Lists. Payment services are another potential area for wireless PKI services to develop.

Certificate distribution can occur in many forms. Device and WIM manufacturers will install client certificates at manufacture and carriers will issue client certificates to their subscriber. Additionally, content (end service) providers may issue client certificates, particularly if they don’t trust the operator to authenticate the user. At the server side, all content providers, as well as potentially mobile operators, and WAP gateway providers, will require server certificates.

Once wireless PKIs are available the potential for applications is enormous. Mobile banking, stock trading and B2C e-Commerce are among the most visible but there are also many other applications. From mobile betting to restaurant, hotel and airline reservations, there are numerous opportunities.

summary

As we have seen, security issues are largely the same whether the environment is mobile and wireless or stationary and tethered. However, there are some additional factors to consider when developing a wireless system. Supplementary security like virtual private networks, and smart cards can help to reduce the vulnerability of mobile devices, and even provide a more secure configuration than is commonly found with current fixed devices.



The graphic above illustrates the interaction of some of the security mechanisms described in this paper. Clearly there are infinitely many possible combinations depending on the network topologies and applications configurations that must be protected. What is important to realize is that it may be relatively simple to implement a secure pilot of one particular technology. However, when it comes to deploying a full set of solutions on a wide scale the challenge grows exponentially if we are to ensure that all fronts are secured but that the user is able to operate without undue effort.

appendix a: related documents

The following key documents and locations provide a wealth of information regarding successful deployments of Wireless Security

Burton White Paper on Wireless Security

http://www.tbg.com/public/doc.asp?docid=244

Bluetooth

http://www.niksula.cs.hut.fi/~jiitv/bluesec.html

http://www.bluetooth.com/developer/download/download.asp?doc=174

Smart Cards

Rankl, Wolfgang & Effing, Wolfgang. **Handbuch der Chipkarten**. Hanser Verlag 1999. ISBN 3-446-21115-2.

PC/SC: http://www.smartcardsys.com/

PKCS#11: http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-11.html

Opencard: http://www.opencard.org/

Javacard: http://www.javasoft.com/products/javacard/index.html

Notice

The information in this publication is confidential and proprietary to Hewlett-Packard Company (HP) and Compaq Information Technologies Group L.P. (Compaq) and is protected by the terms of an end-user license agreement. The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it enable compliance with any federal state or local requirements.

Feedback may be addressed directly to john.rhoton@compaq.com

invent

Technical information contained in this document is subject to change without notice.

5/2002